Learning Maya

Learning Maya 2

Education Publishing Group:

Steve Christov, Deion Green, Bob Gundu, Robert Magee, Elizabeth Mastrotucci, Carla Sharkey

Special thanks to:

Padmé Amidala, Don Chong, Bruce Darrell, Dallas Good, Shawn Dunn, Haskell Friedman, Alan Harris, Steve Hinan, Lincoln Holme, Jill Jacob, Lisa Kelly, Nicholas Li, Cory Mogk, Remko Noteboom, Alan Opler, Lorna Saunders, Jason Ungerman

The following are trademarks of Alias|Wavefront, a division of Silicon Graphics Limited:

Advanced Visualizer™
Alias®
Alias Metamorph™
Alias MotionSampler™
Alias MultiFlip™
Alias Natural Phenomena™
Alias OpenModel™
Alias OpenRender™
Alias OptiF/X™
Alias OpenRender™
Alias PowerAnimator™
Alias PowerTracer™
Alias QuickRender™
Alias QuickShade™
Alias QuickWire™
Alias RayCasting™
Alias RayTracing™
Alias SDL™
Alias ShapeShifter™
Alias StudioPaint™
Conductors™
Dynamation™
Explore™
Kinemation™
Maya™ Artisan
Maya™ Cloth
Maya™ Complete
Maya™ Fur
Maya™ Fusion
Maya™ F/X
Maya™ Invigorator
Maya™ Live
Maya™ MEL
Maya™ PowerModeler
Maya™ Unlimited
SuperConductors™
Wavefront™
Wavefront Composer™
Wavefront Composerlite™
Wavefront IPR™
Wavefront MediaStudio™
Wavefront 3Design™
ZaP!iT™

Maya is a trademark of Silicon Graphics, Inc. exclusively used by Alias|Wavefront, a division of Silicon Graphics Limited.

Microsoft and Windows NT are registered trademarks of Microsoft Corporation in the United States and/or other countries. Graph Layout Toolkit Copyright „ 1992-1996 Tom Sawyer Software, Berkeley, California, All Rights Reserved.

All other product names mentioned are trademarks or registered trademarks of their respective holders.

This document contains proprietary and confidential information of Alias|Wavefront, Inc. and is protected by Federal copyright law. The contents of this document may not be disclosed to third parties, translated, copied, or duplicated in any form, in whole or in part, without the express written permission of Alias|Wavefront, Inc.

The information contained in this document is subject to change without notice. Neither Alias|Wavefront, Inc. nor its employees shall be responsible for incidental or consequential damages resulting from the use of this material or liable for technical or editorial omissions made herein.

CONTENTS

LESSON 19: ADDING FACIAL DETAILS........... 375

LESSON 20: ANIMATING SALTY..................... 401

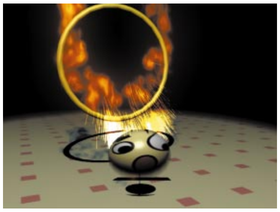
Introduction

Learning Maya 2

Maya is a character animation and visual effects system designed for the professional animator. Built on a procedural architecture called the *Dependency graph*, Maya offers incredible power and flexibility for generating digital images of animated characters and scenes.

This tutorial book gives you hands-on experience with Maya as you complete a series of project-focused lessons. In each project, you will model, animate, texture map, add visual effects and render.

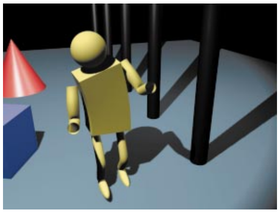
Shown below are the five projects that you will be animating as you complete this book:



Project One: Bouncing Ball

Project Two: Jack-in-the-box

Project Three: Space Battle



Project Four: Primitive Man

WATERWORLD

Project Five: Salty the Seal

How to use this book

How you use this book will depend on your experience with computer graphics and 3D animation. This book moves at a fast pace and is designed to help both the novice and the expert as outlined below:

Novice - If this is your first experience with 3D software, we suggest that you glance over each lesson before you complete it. This will give you a better understanding of what is going to happen before you tackle the software directly.

Intermediate - If you are already familiar with the world of 3D animation, you can dive in and complete the lessons as written. You may want to keep the *Using Maya* manual handy to help give you a more complete understanding of each tool.

Expert - Experts can run through the exercises in this book at a quicker pace. The location of Maya's tools and the terminology surrounding its use will be the most helpful aspect of this book for experts.

Upgrading from 1.0 - If you have used a previous version of Maya then we recommend focusing on projects one and two. Project one is a significant update with useful information while Project two contains the most information about new Version 2 features.

What you should already know

This book assumes that you have some experience with computers and computer graphics. Basic skills such as how to use a three button mouse and the keyboard are essential. Some familiarity with using a computer to draw, paint or model would help.

Computer hardware

Maya is designed to take full advantage of your system's graphics capabilities. This means that if you have a system that supports texture RAM then you can display fully textured models interactively as you work.

The images in this book make use of these graphics capabilities. If you do not have texture RAM then you will have to find alternative methods for previewing textures and lights.

IRIX and Windows NT

This book is written to cover both IRIX and Windows NT platforms. Graphics and text have been modified where applicable. You may notice that your screen varies slightly from the illustrations depending on the platform you are using.

Things to watch for:

- Window focus may differ. For example, if you are on Windows NT, you have to click in the panels with our middle mouse button to make it active.

Maya packaging

This book can be used with either **Maya Complete** or **Maya Unlimited** . It's focus is on functionality that is shared between the two packages. These include all the tools necessary for modeling, animating and rendering a scene, as well as, particle and dynamic effects and Maya's Artisan tools.

Installing tutorial files

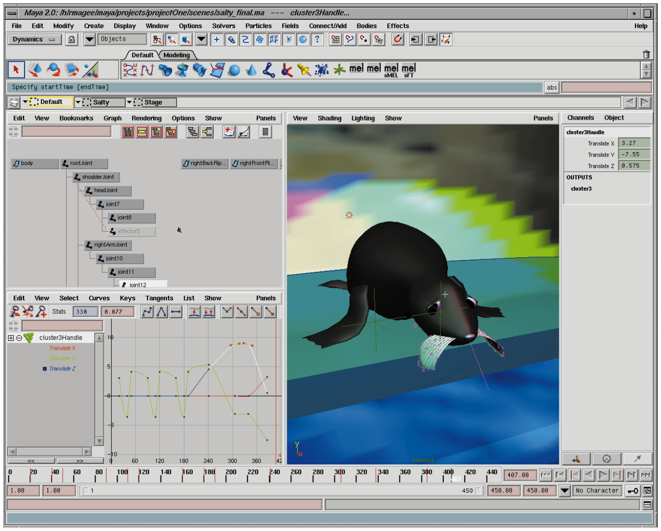
Project directories are available on the *Learning Maya 2 CD-ROM* that you can find at the back of this book. Follow the instructions outlined on the CD.

Understanding Maya

To understand Maya, it helps to understand how Maya works at a conceptual level. This introduction is designed to give you the *story* about Maya. This means that the focus of this introduction will be on how different Maya concepts are woven together to create an integrated workspace.

While this book teaches you how to model, animate and render in Maya, these concepts are taught with a particular focus on how Maya's underlying architecture supports the creation of animated sequences.

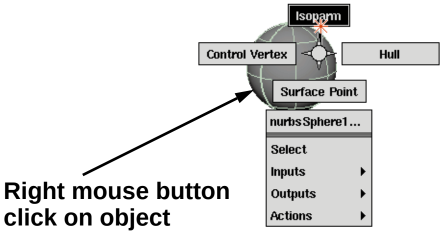
You will soon learn how Maya's architecture can be explained using a single line – *nodes with attributes that are connected*. As you work through this book, the meaning of that statement becomes clearer and you will learn to appreciate how Maya's interface lets you focus on the act of creation, while giving you access to the power inherent in the underlying architecture.


Maya 2.0: /h/rmagee/maya/projects/projectOne/scenes/salty_final.ma --- cluster3Handle...
File Edit Modify Create Display Window Options Solvers Particles Fields Connect/Add Bodies Effects Help
Dynamics
Objects
Default Modeling
Specify startTime [endTime]
Default Salty Stage
Edit View Bookmarks Graph Rendering Options Show Panels
View Shading Lighting Show Panels
Channels Object
cluster3Handle
Translate X 3.27
Translate Y -7.55
Translate Z 0.575
OUTPUTS
cluster3
body rootJoint shoulderJoint headJoint joint7 joint8 effector5 rightArmJoint joint10 joint11 joint12 rightBackFlip... rightFrontFli...
Edit View Select Curves Keys Tangents List Show Panels
Stats 330 8.877
cluster3Handle
Translate X Translate Y Translate Z
No Character

A typical Maya workspace layout

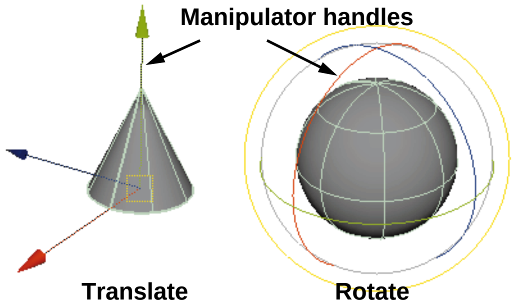
The user interface

The Maya user interface includes a number of tools, editors and controls. You can access these using the main menus or using special context-sensitive marking menus. You can also use *shelves* to store important icons or hotkeys to speed up workflow. Maya is designed to let you configure the user interface as you see fit.



Marking menu

To work with objects, you can enter values using coordinate entry or you can use more interactive 3D manipulators. Manipulator handles let you edit your objects with a simple click-drag.



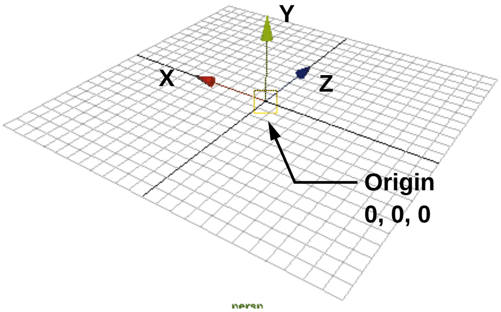
Maya manipulators

Maya's user interface supports multiple levels of *undo* and *redo* and includes a drag-and-drop paradigm for accessing many parts of the workspace.

You will learn how to use the various user interface elements in **Lesson One** and the **Lesson 5**.

Working in 3D

In Maya, you will build and animate objects in three dimensions. These dimensions are defined by the cardinal axes which are labeled as X, Y and Z. These represent the length (X), height (Y) and depth (Z) of your scene. These axes are represented by colors – red for X, green for Y and blue for Z.



The cardinal axes

In Maya, the Y-axis is pointing up which is also referred to as *Y-up*.

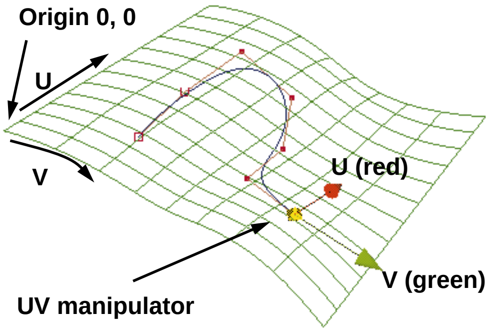
As you position, scale and rotate your objects, these three axes will serve as your main points of reference. The center of this coordinate system is called the origin and has a value of 0, 0, 0.

UV coordinate space

As you build surfaces in Maya, they are created with their own coordinate space that is define by U in one direction and V in another. You can use these coordinates when you are working with *curve on surface* objects or when you are positioning textures on a surface.

One corner of the surface acts as the origin of the system and all coordinates lie directly on the surface.

You can make surfaces *live* in order to work directly in the UV coordinate space. You will also encounter U and V attributes when you place textures onto surfaces.



UV coordinates on a live surface

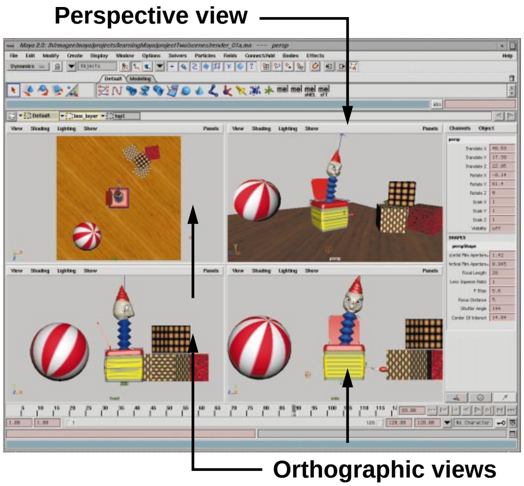
Views

In Maya, you visualize your scenes using view panels that let you see into the 3D world.

Perspective views let you see your scene as if you were looking at it with your own eyes or through the lens of a camera.

Orthographic views are parallel to the scene and offer a more objective view. They focus on two axes at a time and are referred to as the *Top*, *Side* and *Front* views.

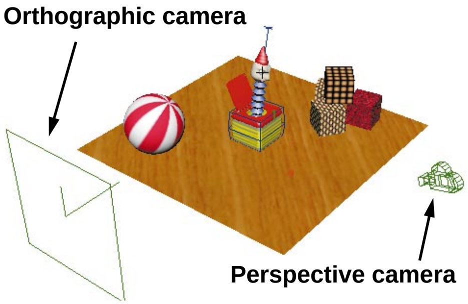
In many cases, you will require several views to help you define the proper location of your objects. An object's position that looks good in the Top view may not make sense in a Side view. Maya lets you view multiple views at one time to help coordinate what you see.



Orthographic and Perspective views

Cameras

To achieve a particular view, you look through a digital camera. An orthographic camera defines the view using a parallel plane and a direction while a perspective camera uses an *eye point*, a *look at point* and a *focal length*.



Perspective and orthographic cameras

You will learn how to use view panels in **Lesson 1** and how to create and animate a camera in **Lesson 17** and **Lesson 21**.

Image planes

When you work with cameras, it is possible to place special backdrop objects called *image planes* onto the camera. An image plane can be placed onto the camera so that as the camera moves, the plane stays aligned.

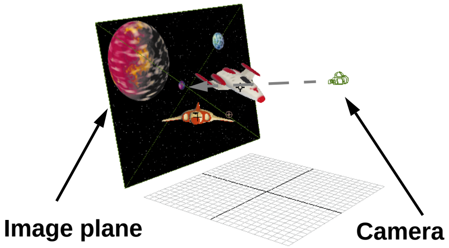


Image plane attached to a camera

The image plane has several attributes that allow you to track and scale the image. These attributes can be animated to give the appearance that the plane is moving.

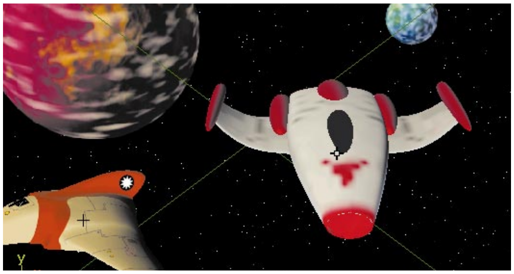


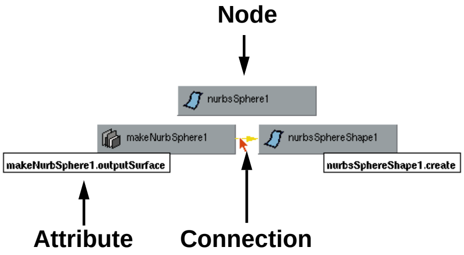
Image plane seen looking through the camera

You will learn how to use image planes in **Lesson 14** as you animate a space backdrop for your battling spaceships.

THE DEPENDENCY GRAPH

Maya's system architecture uses a procedural paradigm that lets you integrate traditional keyframe animation, inverse kinematics, dynamics and scripting on top of a node-based architecture that is called the **Dependency graph**. If you wanted to reduce this graph to its bare essentials, you could describe it as *nodes with attributes that are connected*. This node-based architecture gives Maya its flexible procedural qualities.

Below is a diagram showing a primitive sphere's Dependency graph. A procedural input node defines the shape of the sphere by connecting attributes on each node.



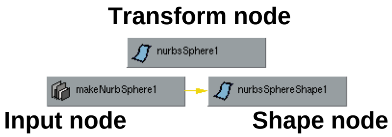
The Dependency graph

Nodes

Every element in Maya, whether it is a curve, surface, deformer, light, texture, expression, modeling operation or animation curve, is described by either a single node or a series of connected nodes.

A *node* is a generic object type in Maya. Different nodes are designed with specific attributes so that the node can accomplish a specific task. Nodes define all object types in Maya including geometry, shading, and lighting.

Shown below are three typical node types as they appear on a primitive sphere.



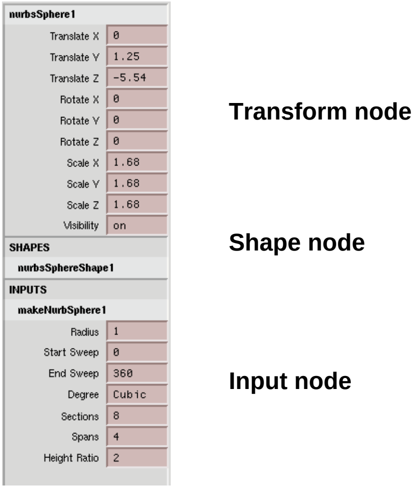
Node types on a sphere

Transform node - Transform nodes contain positioning information for your objects. When you move, rotate or scale, this is the node you are affecting.

Input node - The input node represents options that drive the creation of your sphere's shape such as radius or endsweep.

Shape node - The shape node contains all the component information that represents the actual look of the sphere.

Maya's user interface presents these nodes to you in many ways. Below is an image of the Channel box where you can edit and animate node attributes.



nurbsSphere1

Translate X	0
Translate Y	1.25
Translate Z	-5.54
Rotate X	0
Rotate Y	0
Rotate Z	0
Scale X	1.68
Scale Y	1.68
Scale Z	1.68
Visibility	on

SHAPES

nurbsSphereShape1

INPUTS

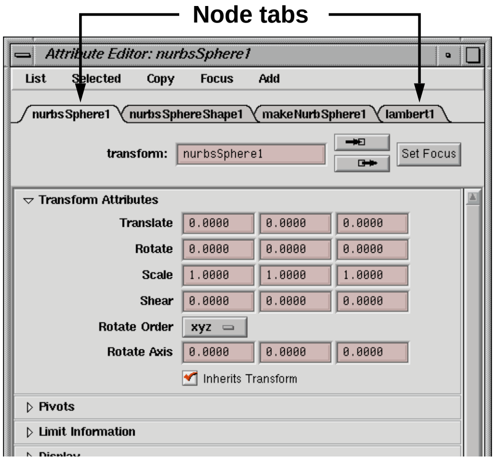
makeNurbSphere1

Radius	1
Start Sweep	0
End Sweep	360
Degree	Cubic
Sections	8
Spans	4
Height Ratio	2

Channel box

Attributes

Each node is defined by a series of attributes that relate to what the node is designed to accomplish. In the case of a transform node, *X Translate* is an attribute. In the case of a shader node, *Color Red* is an attribute. It is possible for you to assign values to the attributes. You can work with attributes in a number of user-interface windows including the *Attribute Editor*, the *Channel box* and the *Spread Sheet Editor*.



transform: nurbsSphere1

Transform Attributes

Translate	0.0000	0.0000	0.0000
Rotate	0.0000	0.0000	0.0000
Scale	1.0000	1.0000	1.0000
Shear	0.0000	0.0000	0.0000
Rotate Order	xyz		
Rotate Axis	0.0000	0.0000	0.0000

☑ Inherits Transform

Attribute Editor

One important feature in Maya is that you can animate virtually every attribute on any node. This helps give Maya its animation power. You should note that attributes are also referred to as *channels*.

Connections

Nodes don't exist in isolation. A finished animation results when you begin making connections between attributes on different nodes. These connections are also known as *dependencies*. In modeling, these connections are sometimes referred to as *construction history*.

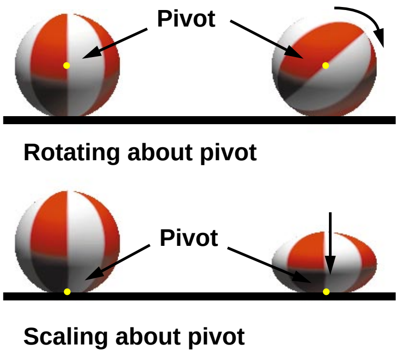
Most of these connections are created automatically by the Maya user-interface as a result of using commands or tools. If you desire, you can also build and edit these connections explicitly using the *Connection editor*, by entering *MEL* (Maya Embedded Language) commands, or by writing MEL-based expressions.

You will take an in-depth look at nodes, attributes and connections in **Lesson 6**.

Pivots

Transform nodes are all built with a special component known as the pivot point. Just like your arm pivots around your elbow, the pivot helps you rotate a transform node. By changing the location of the pivot point, you get different results.

Pivots are basically the stationary point from which you rotate or scale objects. When animating, you sometimes need to build hierarchies where one transform node rotates the object and a second transform node scales. Each node can have its own pivot location to help you get the effect you want.



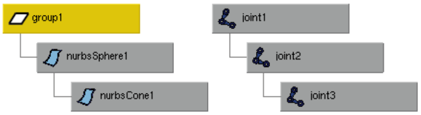
Rotation and scaling pivots

You will be working with transform node pivots in **Lesson 7**.

Hierarchies

When you are building scenes in Maya, you have learned that you can build dependency connections to link node attributes. When working with transform nodes or joint nodes, you can also build hierarchies which create a different kind of relationship between your objects.

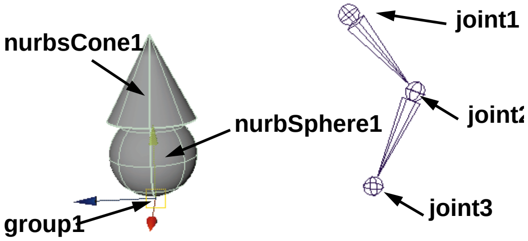
In a hierarchy, one transform node is *parented* to another. When Maya works with these nodes, Maya looks first at the top node, or *root* node, then down the hierarchy. Therefore, motion from the upper nodes is transferred down into the lower nodes. In the diagram below, if the *group1* node is rotated, then the two lower nodes will rotate with it. If the *nurbsCone* node is rotated, the upper nodes are not affected.



Object and joint hierarchy nodes

Joint hierarchies are used when you are building characters. When you create joints, the joint pivots act as limb joints while bones are drawn between them to help visualize the joint chain. By default, these hierarchies work just like object hierarchies. Rotating one node rotates all of the lower nodes at the same time.

You will learn more about joint hierarchies later in this introduction where you will also learn how *inverse kinematics* can reverse the flow of the hierarchy.

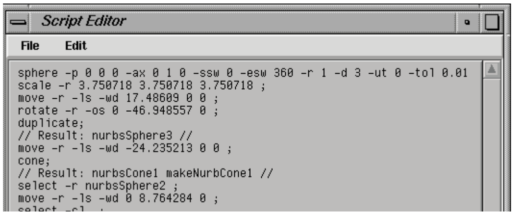


Object and joint hierarchies

You will be working with hierarchies in **Lesson 6**. You will work with joint hierarchies in **Lesson 9**, and **Lesson 16**.

MEL scripting

MEL stands for Maya Embedded Language. In Maya, every time you use a tool or open a window, you are using MEL. MEL can be used to execute simple commands, write expressions or build scripts that will extend Maya's existing functionality.



```
sphere -p 0 0 0 -ax 0 1 0 -ssw 0 -esw 360 -r 1 -d 3 -ut 0 -tol 0.01
scale -r 3.750718 3.750718 3.750718 ;
move -r -ls -wd 17.48609 0 0 ;
rotate -r -os 0 -46.948557 0 ;
duplicate;
// Result: nurbsSphere3 //
move -r -ls -wd -24.235213 0 0 ;
cone;
// Result: nurbsCone1 makeNurbCone1 //
select -r nurbsSphere2 ;
move -r -ls -wd 0 8.764284 0 ;
select -cl ;
```

The Windows NT Script Editor

MEL is the perfect tool for technical directors who are looking for easy access to customizing how Maya works to suit the needs of a particular production environment. Animators can also use MEL to create simple macros that will help speed up more difficult or tedious workflows.

You will learn how to work with MEL to build custom procedures in **Lesson 23**.

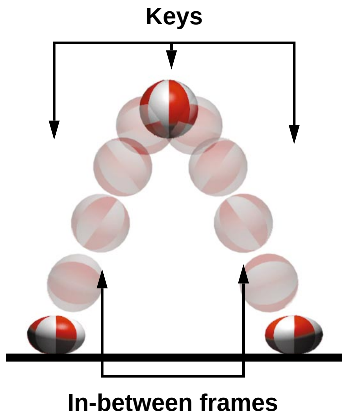
ANIMATING IN MAYA

When you animate, you bring objects to life. In Maya, there are several different ways in which you can animate your scenes and the characters who inhabit them.

Animation in Maya is generally measured using frames that mimic the frames you would find on a film reel. You can play these frames at different speeds to achieve an animated effect. By default, Maya plays at 24 frames for every second.

Keyframe animation

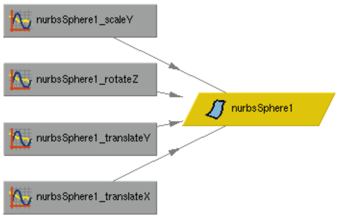
The most familiar method of animating is called *keyframe animation*. Using this technique, you determine how you want the parts of your objects to look at a particular frame, then you save the important attributes as keys. After you set several keys, the animation can be played back with Maya filling motion in between the keys.



Keys and in-between frames

When keys are set on a particular attribute, the keyed values are stored in special nodes called *animation curves* nodes.

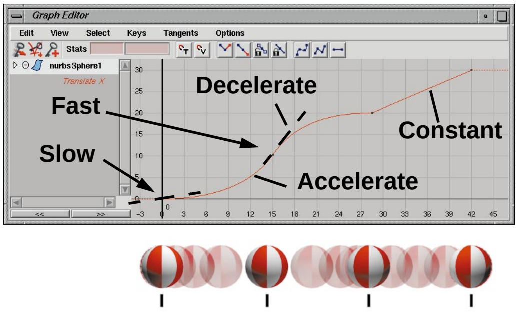
These curves are defined by the keys which map the value of the attribute against time. The following is an example of several animation curve nodes connected to a transformation node. One node is created for every attribute that is animated.



Dependency graph showing curve nodes

Once you have a curve, you can begin to control the tangency at each key to determine the quality of the motion in between the main keys. You can make your objects speed up or slow down by editing the shape of these animation curves.

Generally, the slope of the graph curve tells you the speed of the motion. A steep slope in the curve means fast motion while a flat curve means no motion. Think of a skier going down a hill. Steep slopes increase speed while flatter sections slow things down.



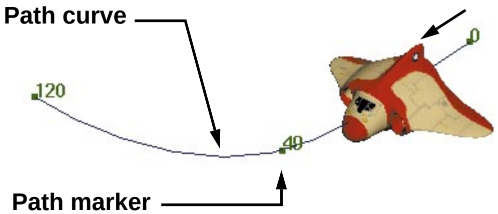
Graph Editor

You will learn how to set keys in **Lesson 1** and how to edit the animation curves in **Lesson 2**. You will then continue to set keys throughout the rest of *Learning Maya*.

Path animation

Path animation is already defined by its name. You can assign one or more objects so that they move along a path which has been drawn as a curve in 3D space. You can then use the shape of the curve and special path markers to edit and tweak the resulting motion.

You will learn how to set up and edit path animations in **Lesson 14**.



Path animation

Reactive animation

Reactive animation is a term used to describe animation in which one object's animation is based on the animation of another object.

An example of this technique would be moving gears where the rotation of one gear is linked to the rotation of other gears. You can then set keys on the first gear and all the others animate automatically. Later, when you want to edit or tweak the keys, only one object needs to be worked on and the others update reactively.

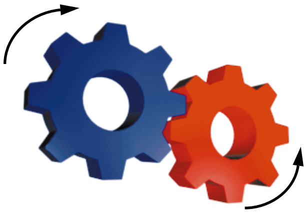


Diagram of animated gears

In Maya, you can set up reactive animation using a number of tools including those outlined below:

Set Driven Key

This tool lets you interactively set up an attribute on one object to drive one or more attributes onto another.

You will learn how to use Set Driven Key in **Lesson 2, Lesson 9** and **Lesson 19**.

Expressions

Expressions are scripts that let you connect different attributes on different nodes.

You will learn how to create expressions in **Lesson 4** and **Lesson 16**.

Constraints

Constraints let you set up an object to *point at*, *orient to* or *look at* another object.

You will learn how to use constraints in **Lesson 9** and **Lesson 16**.

Connections

Attributes can be directly linked to another attribute using dependency node connections. You can create this kind of direct connection using the Connection Editor.

You will learn how to make these kinds of connections in **Lesson 6**

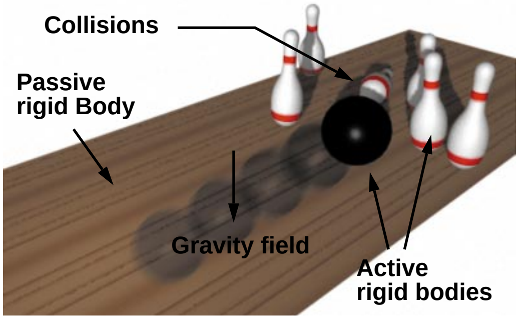
Dynamics

Another animation technique is *dynamics*. You can set up objects in your Maya scene that animate based on physical effects such as collisions, gravity and wind. You set up the different variables such as *bounciness, friction* or *initial velocity*. When you playback the scene, you run a simulation to see how all the parts react to the variables.

This technique gives you natural motion that would be difficult to keyframe. You can use dynamics with rigid body objects, particles or soft body objects.

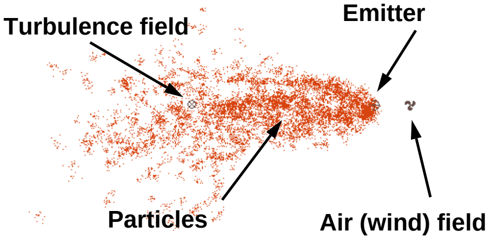
Rigid body objects are objects that don't need to be deformed. You set up this kind of simulation by setting objects as either *active* or *passive rigid bodies*. Active bodies react to the dynamics, whereas passive bodies act only as colliding objects for active bodies.

To simulate effects such as wind or gravity, you use dynamic *fields* that are added to the scene then connected to your objects.



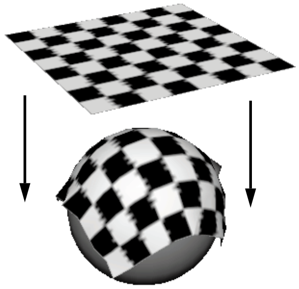
Rigid body simulation of bowling ball and pins

Particles are tiny points that can be used to create effects such as smoke, fire or explosions. These points are emitted into the scene where they are also affected by the dynamic fields.



Particles

Soft bodies are surfaces that you want to deform during a simulation. To create a soft body, you will combine a surface with a series of particles. The particles react to the dynamic forces to help make the surface deform.



Soft bodies

You will learn how to use particle dynamics in **Lesson 4** and **Lesson 15**.

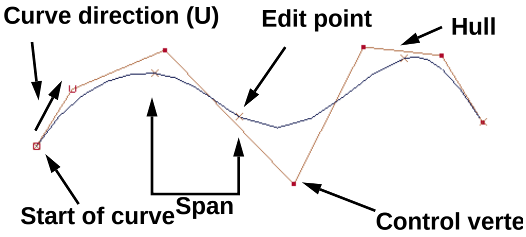
MODELING IN MAYA

The objects you want to animate in Maya are usually built using either NURBS surfaces or polygonal meshes. Maya offers you both of these geometry types so that you can choose the method best suited for your work.

NURBS curves

NURBS stands for *non-uniform rational b-splines* which is a technical term for a spline curve. By modeling with NURBS curves, you can lay down control points and smooth geometry will be created using the points as guides.

Shown below is a typical NURBS curve with important parts labelled:



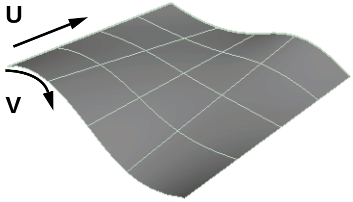
A NURBS curve

These key components define important aspects of how a curve works. The flexibility and power of NURBS geometry comes from your ability to edit the shape of the geometry using these controls.

As your geometry becomes more complex, you get more of these controls. For this reason, it is usually better to build using simpler geometry so that you can more easily control the shape. If you need more complex geometry, then controls can be inserted later.

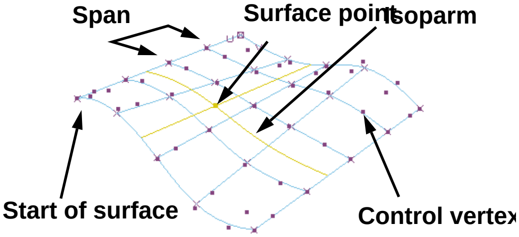
NURBS surfaces

Surfaces are defined using the same mathematics as curves except now there are in two dimensions – U and V. You learned about this earlier when you learned about UV coordinate space.



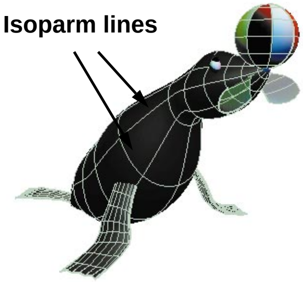
A NURBS surface

Below are some of the component elements of a typical NURBS surface:



NURBS components

Complex shapes can be, in essence, sculpted using this surface type as you push and pull the controls to shape the surface.



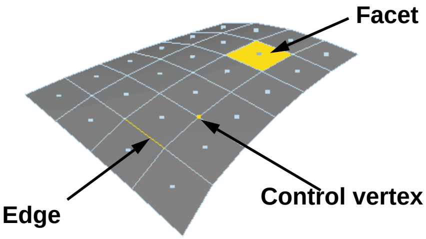
A completed NURBS model

You will learn how to model with NURBS geometry in **Lesson 7, Lesson 13** and **Lesson 18**.

Polygons

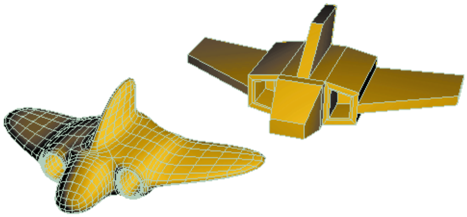
Polygons are another geometry type available in Maya. Whereas NURBS surfaces interpolate the shape of the geometry interactively, polygonal meshes draw the geometry directly to the control vertices. When a polygonal mesh is rendered, it is then interpolated to see a smoother form.

Below are some of the components found on a polygonal mesh:



Polygon components

You can build up polymeshes by extruding, scaling and positioning polygonal facets to build shapes. You can then smooth the shape to get a more organic look for your model.



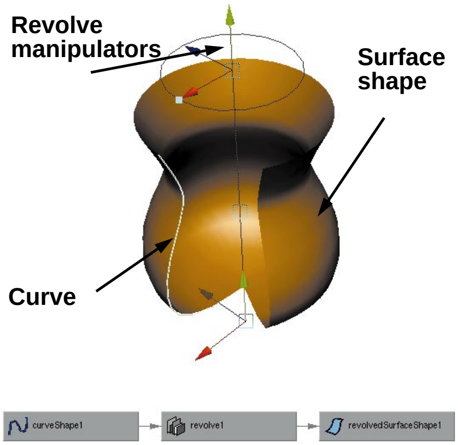
A polygonal model before and after smoothing

You will learn how to build a polygonal model in **Lesson 12**.

Construction history

When you create models in Maya, the various steps are recorded as dependency nodes that remain connected to your surface.

In the example below, a curve has been used to create a revolved surface. Maya keeps the history by creating dependencies between the curve, a revolve node and the shape node. Edits made to the curve or the revolve node will update the final shape.



Revolve surface with dependencies

Many of these nodes come with special manipulators that make it easier to update the node attributes. In the case of the revolve, manipulators are available for the axis line and for the revolve's sweep angle.

It is possible to later delete history so that you are only working with the shape node. Don't forget though, that the dependency nodes have attributes that can be animated. Therefore, you lose some power if you delete history.

You will learn how to work with history in **Lesson 6**, **Lesson 12** and **Lesson 13**.

DEFORMATIONS

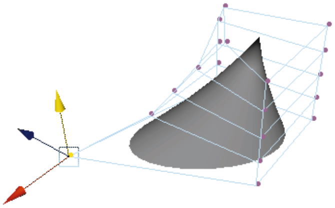
Deformers are special object types that can be used to reshape other objects. By using deformers, you can give your animations more of a squash and stretch quality.

A powerful feature of Maya's deformers is that they can be layered for more subtle effects. You can also bind deformers into skeletons or affect them with soft body dynamics.

The following lists some of the key deformer types available in Maya.

Lattices

Lattices are external frames that can be applied to your objects. If you then reshape the frame the object is deformed in response.

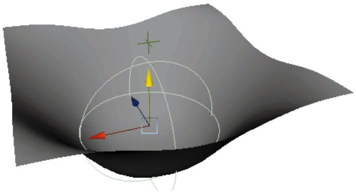


Lattice deformer

You will learn how to work with lattices in **Lesson 10, Lesson 12, Lesson 16** and**Lesson 19.**

Sculpt objects

Sculpt objects let you deform a surface by pushing it with the object. By animating the position of the sculpt object, you can achieve animated surface deformations.

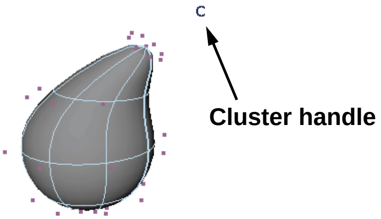


Sculpt object deformer

You will learn how to work with a sculpt flexor in **Lesson 16.**

Clusters

Clusters are groups of CVs or lattice points that are built into a single set. The cluster is given its own pivot point and can be used to manipulate the clustered points. You can weight the CVs in a cluster to get more subtle effects.



Cluster deformer

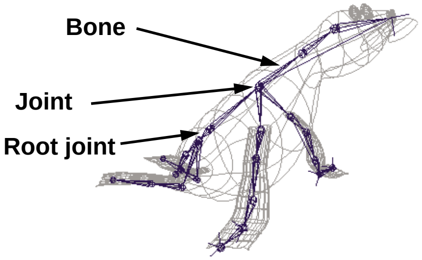
You will learn how to work with clusters in **Lesson 20.**

CHARACTER ANIMATION

In Maya, character animation typically involves the animation of surfaces using skeleton joint chains and inverse kinematic handles to help drive the motion. At the same time, the chains can be set up to work with special sculpt objects and lattices known as *flexors*. These let you perform surface deformations that will add realism to your character.

Skeletons and joints

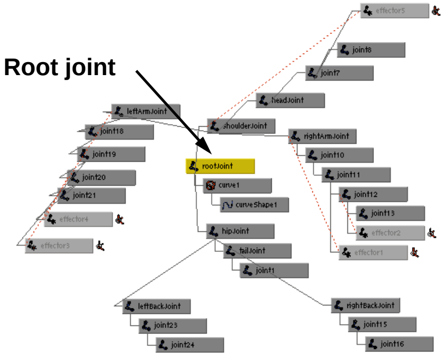
As you have already learned, skeleton joint chains are actually hierarchies. A skeleton is made up of joint nodes that are connected visually by bone icons. These hierarchies let you group or bind in geometry which creates your surface deformations.



Joints and bones

Inverse kinematics

By default, joint chains work like any other hierarchy. The rotation of one joint is transferred to the lower joint nodes. This is known as *forward kinematics*. While this method is powerful, it makes it hard to plant a character's feet or move a hand to control the arm.



Character joint hierarchy

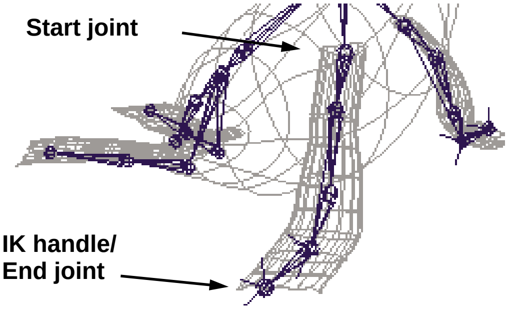
Inverse kinematics lets you work with the hierarchy in the opposite direction. By placing an IK handle that runs from a start joint to an end joint, you can better control the chain of joints. There are three kinds of solvers in Maya – the IK spline, the IK single chain and the IK rotate plane.

Each of these solvers is designed to help you control the joint rotations using the IK handle as a goal. As the IK handle moves, the IK solver defines the joint rotations that let the end joint move to the IK handle position.

The individual solvers have their own unique controls. Some of these are outlined below:

Single chain solver

The *single chain solver* provides a straight forward mechanism for posing and animating a chain. By moving the IK handle, the chain will update so that the joints lie on a plane.

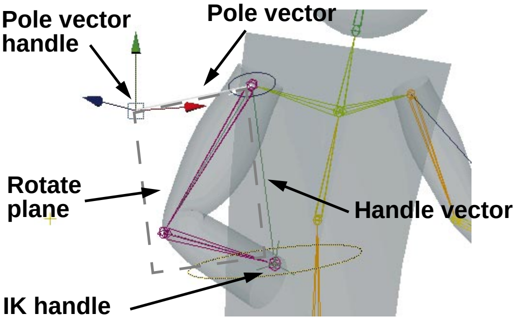


IK single chain solver

You will learn how to use the IK single chain solver in **Lesson 20**.

Rotate plane solver

The *rotate plane solver* gives you more control. With this solver, the plane that acts as the goal for all the joints can be moved by rotating the plane using a *twist attribute* or by moving the *pole vector handle*.

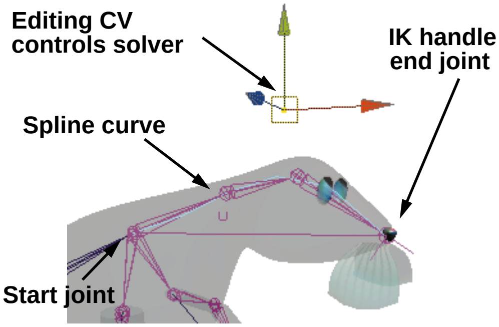


IK rotate plane solver

You will learn how to use the IK rotate plane solver in **Lesson 16**.

IK spline solver

The IK spline solver lets you control the chain using a spline curve. You can edit the CVs on the spline to update the rotation of the joints in the chain.



IK spline solver

You will learn how to use the IK spline solver in **Lesson 9** and **Lesson 20**.

Skinning your characters

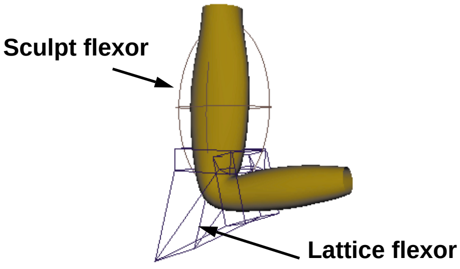
Once you have a skeleton built, you can bind skin the surfaces of your character so that they deform with the rotation of the joints. In Maya you can use either soft skinning or hard skinning. Soft skinning uses weighted clusters while hard skinning does not.

Surface deformations

You will learn how to use soft skinning in **Lesson 9** and hard skinning in **Lesson 18**.

Flexors

In many cases, skinning a character does not yield realistic deformations in the character's joint areas. You can use *flexors* to add this secondary level of deformations to help with the tucking and bulging of your character.



Flexors

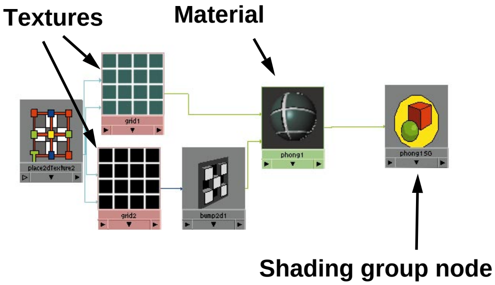
You will learn how to work with flexors in **Lesson 16**.

RENDERING

Once your characters are set up, you can apply color and texture, then render with realistic lighting.

Shading groups

In Maya, you add texture maps and other rendering effects using *shading groups*. A shading group is a network of dependency nodes that all connect into a shading group node. Even the assigned surfaces and related lights are part of a shading group.



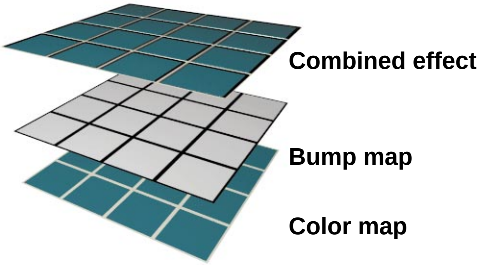
Shading group dependencies

You can think of a shading group as a sort of bucket into which you place all the color, texture and material qualities that you want on your surface. You then dip in the surface and throw in a light or two and the final effect is achieved.

You will learn how to create shading groups in **Lesson 3**, **Lesson 11**, **Lesson 12 Lesson 13** and **Lesson 21**.

Texture maps

To add detail to your shading groups, you can *texture map* different attributes. Some of these include bump, transparency and color.

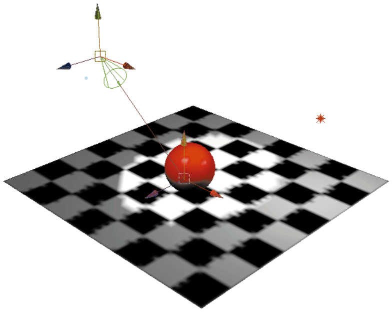


Texture map layers

You will learn how to texture map objects in **Lesson 3** and **Lesson 11**. Other mapping techniques will be used in **Lesson 12** and **Lesson 13**.

Lighting

Before you render, you can light your scenes using any number of lights. These lights let you add mood and atmosphere to a scene in much the same way as lighting is used by a photographer. Maya lets you preview your lights interactively as you model, or you can render to see the final effect.

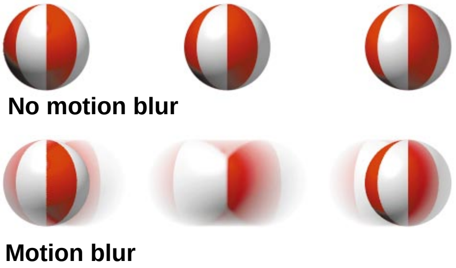


Light manipulator

You will start applying lights in **Lesson 3**.

Motion blur

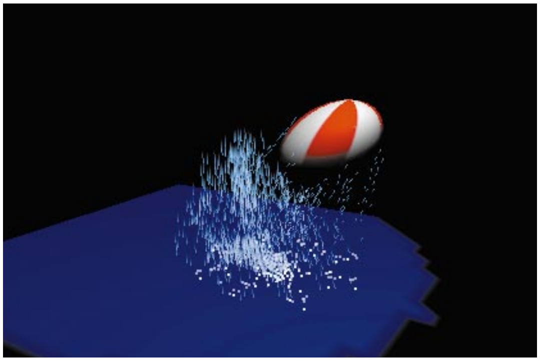
When a real-life camera takes a shot of a moving object the final image is often blurred. This *motion blur* adds to the animated look of a scene and can be used in Maya. Maya contains two types of motion blur – a 2 1/2 D solution and a 3D solution. You will use both these types of motion blur in this book.



Motion blur

Hardware rendering

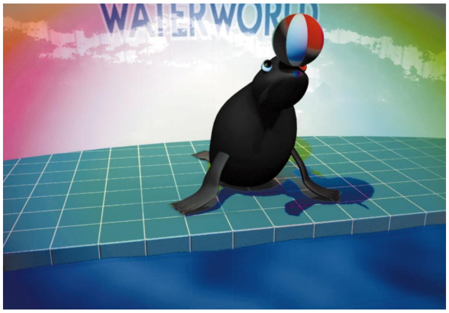
Maya includes *hardware rendering* that lets you see rendered images on your screen for previewing your animations. You can also use the hardware renderer to render some particle effects. These effects can later be composited in with software rendered images of your geometry.



Hardware rendering

A-buffer rendering

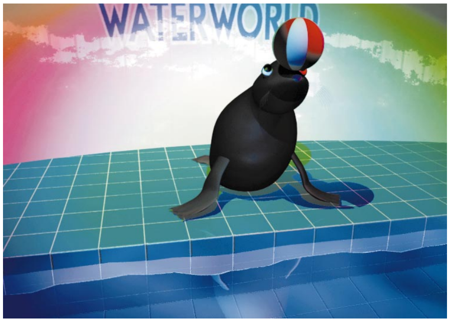
By default, Maya uses an *A-buffer* renderer for *software rendering*. This render type lets you see shaded scenes with shadows and motion blur. This method of rendering is great for most of your rendering needs. If you want reflections and refractions, then you need to turn on raytracing.



A-buffer rendering

Raytrace rendering

Raytracing lets you include reflections and refractions into your scenes. Maya has a selective raytracer which means that only objects or shading groups that have raytrace capabilities will use this renderer. Raytracing is slower than the default A-buffer and should be used only when it is required to enhance the look of a scene.

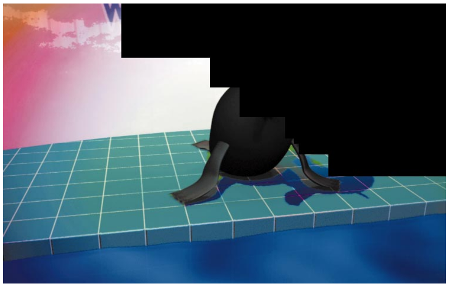


Raytrace rendering

How the renderer works

The Maya renderer works by looking through the camera at the scene. It then takes a section or tile, and analyzes whether or not it can render that section. If yes, then it combines the information found in the shading group (geometry, lights and shading network) with the Render Globals information, and the whole tile is rendered.

As the renderer moves on to the next section, it again analyzes the situation. If it hits a tile where there is more information than it wants to handle at one time, it then breaks the tile down into a smaller tile and renders.



Rendering of A-buffer tiles in progress

When you use raytracing, each tile is first rendered with the A-buffer, then the renderer looks for items that require raytracing. If it finds any, then it layers in the raytraced sections. When it finishes, you have your finished image, or if you are rendering an animation, a sequence of images.

You will render scenes in **Lesson 3**, **Lesson 11**, **Lesson 15**, **Lesson 17** and **Lesson 22**.

IPR

Maya includes a new Interactive Photorealistic Renderer that gives you fast feedback for texturing and lighting updates. You will use IPR in **Lesson 11**.

Conclusion

Now that you have a basic understanding of what Maya is designed to do, it is time for you to start working with the system directly. The concepts outlined in this introduction will make a lot more sense when you experience them first hand.

Don't forget to place the *learningMaya* directory into your project directory so that you have tutorial support files on hand. If you haven't done this yet, you can still complete the **Getting Started** section which does not require any support files.

Lesson 1

Bouncing the Ball

Lesson 2

Adding Character

Lesson 3

Rendering

Lesson 4

Particles and Dynamics

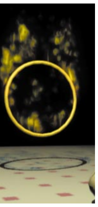
Lesson 5

Working with Maya

Lesson 6

The Dependency Graph

Project One

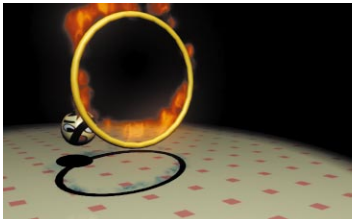


Project One

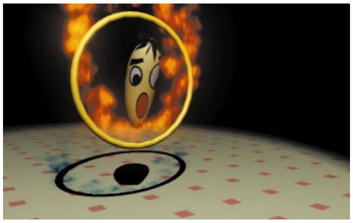
In Project One, you are going to animate a ball bouncing through a ring of fire. This will give you the chance to work with the Maya workspace while animating a scene.

You will learn about building models, adding textures and light, keyframing your objects and rendering your scene. With the ring of fire, you will also explore Maya's preset particle effects while adding your own particles to create the sparks that are emitted when the ball hits the flames.

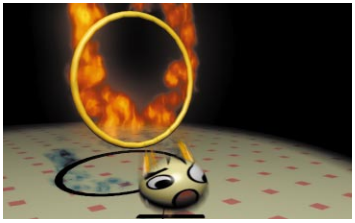
After bouncing the ball, you will then take a more in-depth look at Maya's user interface and the dependency graph. These lessons offer you a more complete look at some of the key concepts and workflows that drive how Maya works.



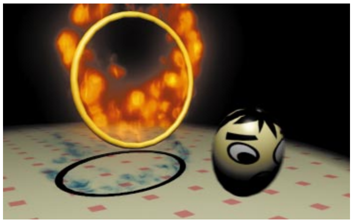
1. The ball begins to bounce towards a flaming ring of fire.



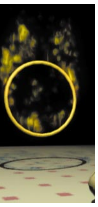
2. Bravely, the ball leaps up into the flames.



3. Sparks fly as the ball brushes past the deadly fire.

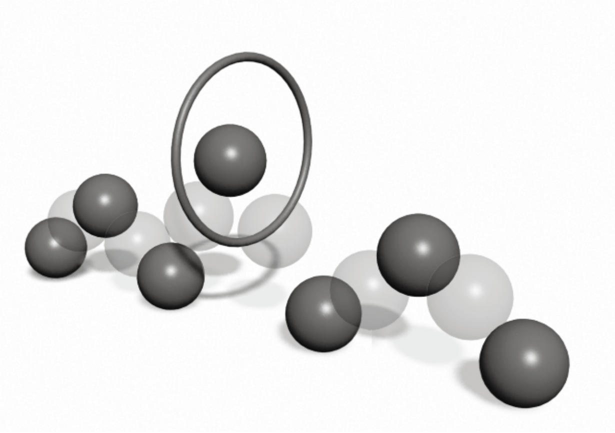


4. Everyone is relieved as the ball bounces off safely with barely a scratch.



1 Bouncing a Ball

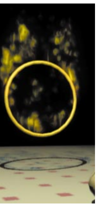
This lesson teaches you how to build and animate a simple bouncing ball that jumps through a hoop. You will explore Maya's user interface as you learn how to build and develop an animated scene.



A bouncing ball

In this lesson you will learn the following:

- How to create primitive objects
- How to move objects in 3D space
- How to use Maya's view tools
- How to change the display of your objects
- How to set keys on your objects
- How to playback animated sequences



Setting up Maya

Before beginning, install Maya 2.0, and add the Learning Maya support files to your projects directory. These can be found on the *Learning Maya 2* CD_ROM.

If you have been working with Maya 2.0 and have changed any of your user interface settings, you may want to delete your preferences in order to start with the default Maya configuration. To do this while saving your existing preferences, see *Appendix A: File Management*.

If you are running multiple versions (v1, v1.5 etc.) of Maya on the same machine, you should also backup and delete the preference files for those versions. Otherwise, Maya 2.0 will use these preferences when it first launches and some of the new marking menu and hotkey settings will not be available. You can replace them later when you know that your Maya 2.0 prefs have been created.

Creating a new project

Maya uses the concept of a project to organize files. You can create project directories that contain sub-directories for storing your files. You will be saving your work into scene files.

1 Launch Maya

To get started, you need to launch Maya:

- Double-click the Maya icon.

Or

- Enter *maya* in a shell window in IRIX or in the command window in Windows NT.

2 Set the courseware project

To manage your files, you can set a project directory that contains sub-directories for different types of files that relate to your project.

- Go to the **File** menu and select **Project → Set...**

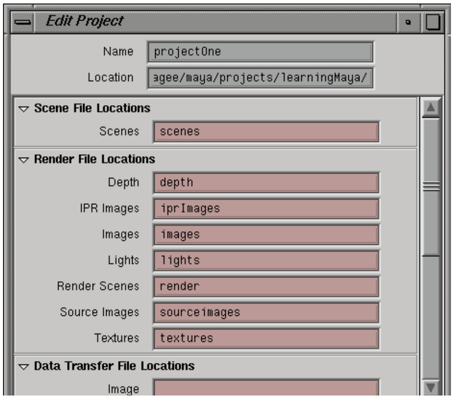
  A window opens that points you to the Maya projects directory.

- Open the folder named *learningMaya*.
- Click on the folder named *projectOne* to select it.
- Click on the **OK** button.

  This sets the learningMaya directory as your current project.

- Go to the **File** menu and select **Project → Edit Current...**

  Make sure that the project directories are set up as shown below. This ensures that Maya is looking into the proper subdirectories when it opens up scene files.



Edit Project

Name: projectOne
Location: agee/maya/projects/learningMaya/

Scene File Locations
Scenes: scenes

Render File Locations
Depth: depth
IPR Images: iprImages
Images: images
Lights: lights
Render Scenes: render
Source Images: sourceimages
Textures: textures

Data Transfer File Locations
Image:

Edit Current Project window

- If any of the entries are incorrect, edit them to match the preceding image shown above.

3 Make a new scene

- Select **File → New Scene.**

  This makes sure that your current scene is part of the new project.

Tip: If you would like to learn more about how Maya organizes files and directories, see *Appendix A: File Management*.

BUILDING OBJECTS

Every scene you create in Maya will begin with objects. These objects can include things like surfaces, deformers, skeleton joints or particle emitters. For this scene, you will build a ball and a floor surface.

Creating the ball

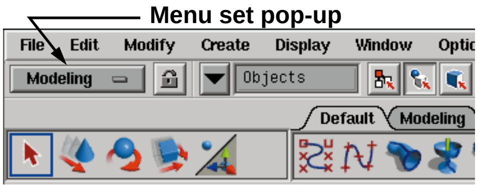
To start, you will build a few simple objects that you will animate, texture and render. The first object is a primitive sphere which will act as the ball.

1 Change menu sets

There are four main menu sets in Maya: *Animation*, *Modeling*, *Dynamics* and *Rendering*. These menu sets are used to access related tool sets.

- From the pop-up menu at the left edge of the Status line, select **Modeling**.

  As you change menu sets, the first seven menus remain the same while the remaining menus change to reflect the chosen menu set.



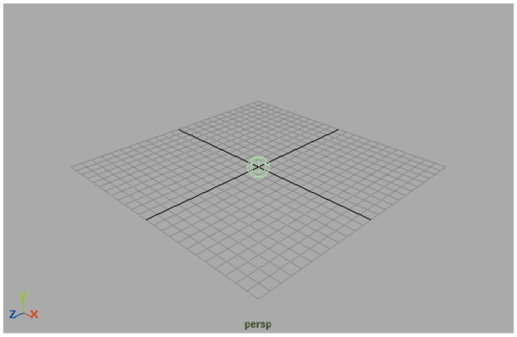
Menu set menu

2 Create a NURBS sphere

A primitive sphere will be used for the ball. It will be built using *non-uniform rational b-spline* (NURBS) geometry. In later lessons, you will learn more about this geometry type.

- From the **Create** menu, select **NURBS Primitives → Sphere.**

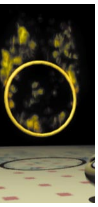
  A sphere is placed at the origin.



Perspective view of sphere

3 Change the ball's radius

The sphere is a procedural model. This means that it is broken down into parts called *nodes*. One node contains its positioning information, one contains its shape information and another contains input information that defines the sphere's construction history using attributes such as radius, sweep, and degree. You can edit



this input node's attributes in the Channel box in order to edit the sphere's shape.

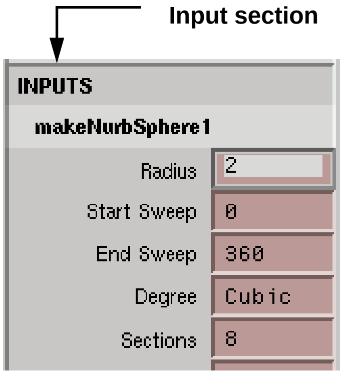
The Channel box is found at the right side of the screen and lets you make changes to key attributes very easily.

- From the channel box's Inputs section, click on *makeNurbSphere*.

  This will make several new attributes available for editing.

- Click on the *Radius* attribute name.
- Type **2** in the entry field then press the **Enter** key.

  Now the sphere is double the size in the perspective view.



INPUTS
makeNurbSphere1

Radius	2
Start Sweep	0
End Sweep	360
Degree	Cubic
Sections	8

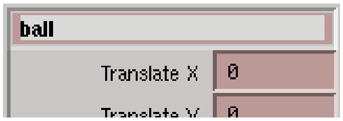
Channel box

Note: Another method for increasing the size of the sphere would be to scale it. In Maya, you can often achieve the same visual results using many different methods. Over time, you will begin to choose the techniques that best suit a particular situation.

4 Rename the ball node

You should rename the existing transform node to make it easier to find later.

- Click on the *nurbsSphere* name at the top of the Channel box to highlight it.
- Type the name *ball* then press the **Enter** key.



ball

Translate X	0
Translate Y	0

Renaming the node in the Channel box

Moving the ball

You will now use the **Move** tool to reposition the ball. This will involve the use of manipulator handles that let you control where you move your object.

1 Position the ball

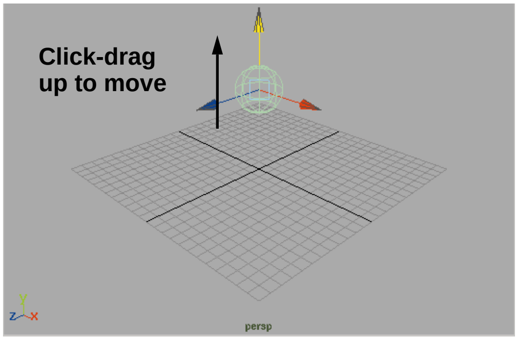
You can now use the **Move** tool to reposition the sphere above the working grid.

- Select the **Move** tool.

  A transform manipulator appears centered on the object.

- Click-drag on the green manipulator handle to move the sphere along the Y-axis.

  The manipulator handle turns yellow to indicate that it is active.



Manipulator handles

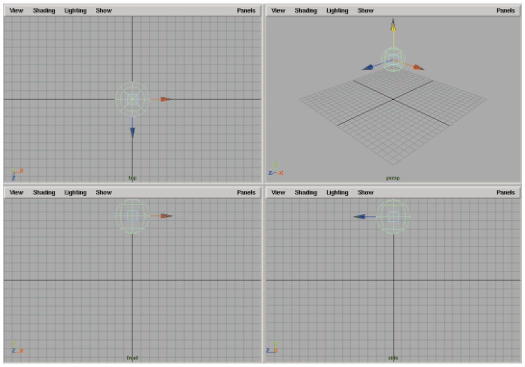
Tip: The transform manipulator has three handles which let you constrain your motion along the X, Y, and Z axes. These are labeled using red for the X-axis, green for the Y-axis and blue for the Z-axis. In Maya, the Y-axis points up by default. This means that Maya is "Y-up" by default.

2 Create four view panels

By default, a single perspective window is shown in the workspace. To see other views of the scene, you can change your panel layout.

- At the top of the Perspective view panel, go to the **Panels** menu and select **Layouts → Four.**

  You can now see the sphere using three orthographic views – Top, Side, and Front – which show you the model from a projected view. You can also see it in a Perspective view that is more like the 3D world we see every day. This multiple view set-up is very useful when positioning objects in 3D space.

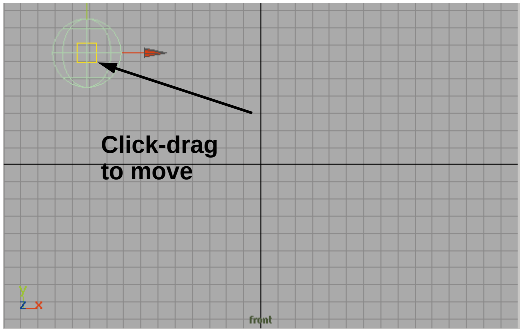


Four view panels

3 Reposition the ball

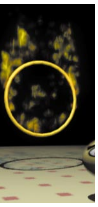
When moving in an orthographic view, you can work in two axes at once by dragging on the center of the manipulator or constraining the motion along a single axis using the handles.

- In the Front view, click-drag on the square center of the manipulator to move the sphere along both the X and the Y axes.

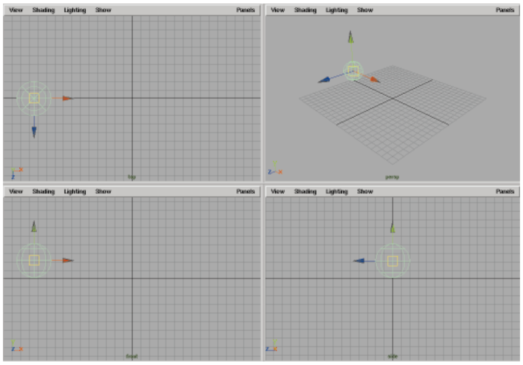


Front view

- Use the manipulator in the various view windows to position the sphere on top of the ground plane as shown below.



Be sure to refer to all four view windows to verify that the object is positioned properly.



Start position of the ball

Note: If you click-drag on the center of the manipulator in the Perspective view, you will notice that it doesn't move along any particular axis. It is actually moving along the camera's view plane.

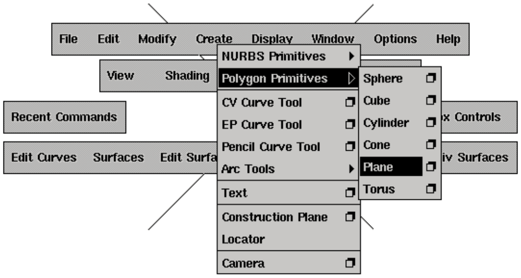
Create a floor surface

To create a floor surface, a primitive plane is used. This plane using polygonal geometry which is different from the sphere which uses NURBS geometry. You can use the hotbox as an alternative method for accessing tools.

1 Create a polygonal plane

- In a view panel, press and hold the spacebar to display the hotbox.
- In the hotbox, select **Create** → **Polygon Primitives** → **Plane.**

  A plane is placed at the origin.



Hotbox access to menu items

Tip: You can access all functions in Maya using either the main menus or the hotbox. As you become more familiar with the hotbox, you can use the user interface options found in the **Options** menu to turn off the panel menus and therefore reduce screen clutter.

2 Change construction history

Just like the sphere, you can edit the plane using its input node. You will increase the size of the plane, then reduce the number of subdivisions to simplify the plane. Since the floor will not be deformed, the extra polygons are not required.

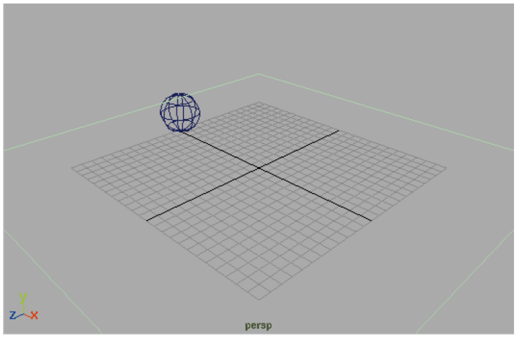
- In the Channel box shown at the right side of your screen, click on the *polyPlane* Input node.
- Set the following:

  Width to **40**;

  Height to **40**;

  Subdivisions X to **1**;

  Subdivisions Y to **1**.



Updated poly plane

3 Rename the plane node

It is a good idea to rename the existing transform node to make it easier to find later.

- Click on the *polyPlane* node's name in the top of the channel box.
- Enter the name *floor*.

Create a Ring

You are now going to add a NURBS torus to the scene that will act as a hoop for the ball to jump through when you animate it bouncing.

1 Create a NURBS Torus

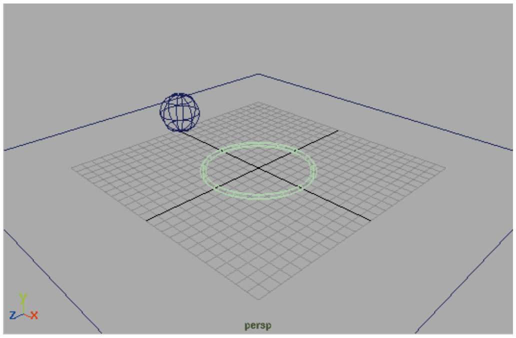
The torus offers a shape that is perfect for the ring. You will create the torus, and then size and position it to get the ring.

- From the **Create** menu, select **NURBS Primitives** → **Torus.**
- In the Channel box, click on the *makeNurbsTorus* Input node.
- Set the following attributes:

  Radius to **5**;

  Sections to **16**;

  Height Ratio to **0.05**.



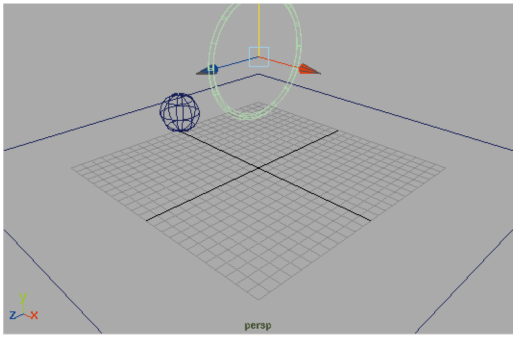
Edited torus

2 Position the Torus

- In the *nurbsTorus* section of the Channel box, set the following attribute:

  Rotate Z to **90**.

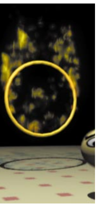
- Rename this node to *ring*.
- With your middle mouse button click in the perspective view then press the **w** key to select the **Move** tool.
- **Move** the ring up so that it sits above the ground plane. Its **Translate Y** attribute should read about **10**.



Ring in place

Viewing the scene

When you work in 3D space, it is important to see your work from different angles. The different view panels let you see your work from the Front, Top, Side and Perspective.



You can also use Maya's view tools to change the views to reposition how you see your scene. In some cases, a view change is like panning a camera around a room, while in other cases a view change might be like turning an object around in your hand to see all the sides. These view tools can be accessed using the **Alt** key in combination with various mouse buttons.

1 Edit the Perspective view

You can use the **Alt** key with either your left mouse button (LMB), your middle mouse button (MMB) or the two together to tumble, track and dolly in your Perspective view.

- Change your view using the following key combinations:

  Alt + LMB to tumble;

  Alt + MMB to track;

  Alt + LMB & **MMB** to dolly.

  You can also use the **Ctrl** key to create a bounding box dolly, where the view adjusts based on a bounding box:

  Ctrl + Alt + LMB to box dolly.

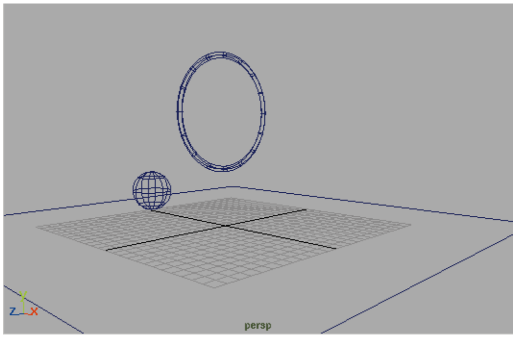
  Click-drag to the right to dolly in and to the left to dolly out.

  You can also undo and redo view changes using the following keys:

  [to undo views; and

  ] to redo views.

- Alter your Perspective window until it appears as shown below.



New view

2 Edit the view in the Side view

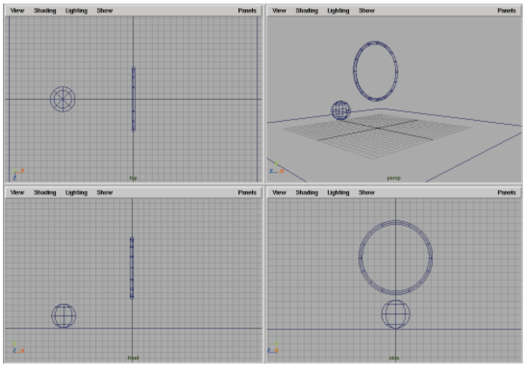
Orthographic views use similar hot keys – except that you cannot tumble an orthographic view.

- In the Side view, change your view using the following key combinations:

  Alt + MMB to track;

  Alt + LMB & **MMB** to dolly

- Keep working with the orthographic views until they are set up as shown in the following.


View Shading Lighting Show Panels
View Shading Lighting Show Panels
View Shading Lighting Show Panels
View Shading Lighting Show Panels
top
persp
front
side

New orthographic views

Setting display options

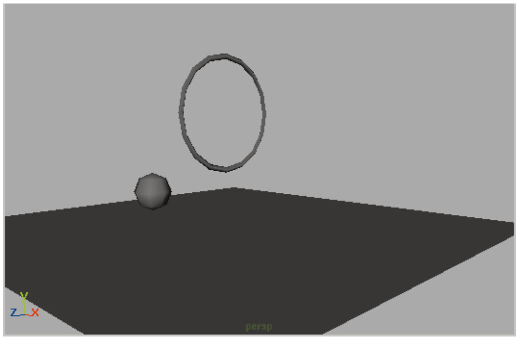
The view panels let you interactively view your scene. By default, this means viewing your scene as a wireframe model. To better evaluate the form of your objects, you can activate hardware shading and increase the sphere's smoothness.

1 Turn on Hardware shading

To help visualize your objects, you can use hardware rendering to display a shaded view within any panel.

- From the Perspective view's **Shading** menu, select **Smooth Shade All.**

  This setting affects all of the objects within the current view panel.



Smooth shaded view

Tip: You can also turn on smooth shading by moving your cursor over the desired panel, clicking with your middle mouse button and pressing the **5** key. The **4** key can be used to return the panel to a wireframe view.

2 Hide the grid

You can hide the grid to simplify your view using one of two options:

- From the Perspective view panel's **Show** menu, select **Grid** to hide the grid for that view only.

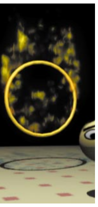
OR

- From the **Display** menu, deselect **Grid** to hide the grid for all views.

3 Adjust the sphere's smoothness

To better evaluate the sphere's surface qualities, you can increase the smoothness.

- **Select** the *ball*.
- From the main **Display** menu, select **NURBS Smoothness** → **Fine.**



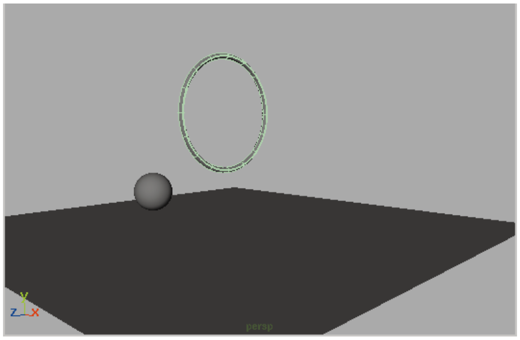
This setting affects how selected NURBS objects are displayed in all view panels.

4 Adjust the ring’s smoothness

Maya also includes some hotkeys for setting surface smoothness.

- Select the *ring*.
- Press the **3** key.

The smoothness setting allows you to see your objects at various degrees of complexity without actually altering the geometry.



New object smoothness

Tip: A NURBS object can have its smoothness set using the following hotkeys:

1 - rough
2 - medium
3 - fine

ANIMATING THE BALL

You now have your scene set up and ready to animate. Using the *ball* node, you will start by keying the overall translation of the ball.

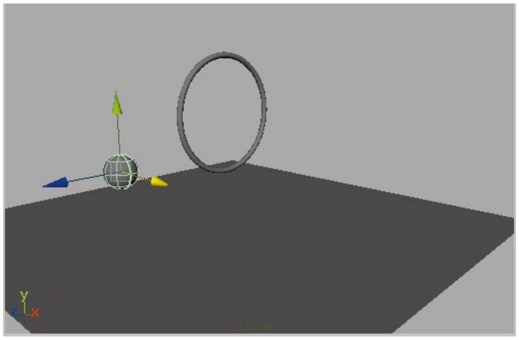
Next, you will add in the peaks and valleys of the bouncing.

Setting keys

To animate the ball, you must define its position at certain points in time. This is accomplished by setting keys. As you playback the animation, Maya reads these key positions and interpolates the position of the ball in between the keyframes.

1 Position the ball

- Select the *ball*.
- **Move** the ball along the X axis to the edge of the *floor*. Use the Red manipulator handle to constrain along X.



Repositioning the ball

2 Set a start key on the ball

You will use the current position to set a key for frame 1.

- **Select** the *ball*.
- Make sure that the current time is set to 1 in the time slider.
- Press **F2** to select the animation menu set.

- From the **Animate** menu, select **Set Key.**

  This places a key at frame 1 for all of the ball's transform attributes.

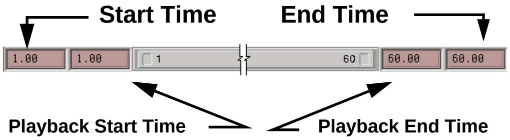
Tip: The menu sets can be changed using the following hotkeys:
F2 - Animation
F3 - Modeling
F4 - Dynamics
F5 - Rendering

---

3 Set end key on the ball

A new position for the ball can be set for frame 60.

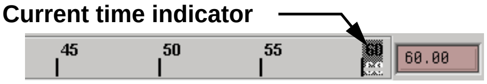
- In the Time slider, change the end time and the playback end time to **60** and press **Enter**.



Time slider frame range settings

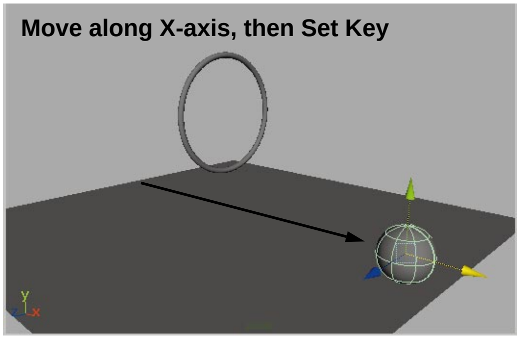
This changes the range of the animation to 60 frames.

- Move the Time slider to frame **60**.



Time slider at frame 60

- Click-drag on the X-axis handle on the sphere to move it forward.

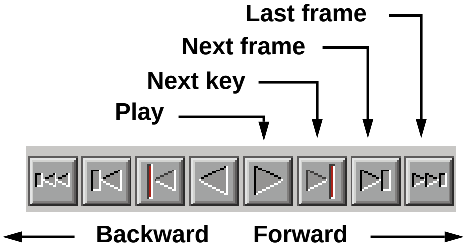


Sphere moved along X-axis

- Press the **s** key to **Set Key** at this position and time.

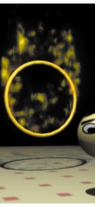
Tip: The **s** key is a hotkey for the Set key command.

- Playback the results using the Playback controls which are located in the lower right corner.



Playback controls

Note: Playback will only be visible in the view panel that is highlighted. If you are not seeing any playback then be sure to click in the perspective panel with your middle mouse button.



4 Add intermediate keys

To make the ball hit the ground four times, you will need two more key positions set at frames 20 and 40.

Any time you want to make sure an object is at a certain place at a certain time, set a key.

- Drag the Time slider to frame **20**.

  The ball is positioned based on existing keys.
- Press the **s** key to **Set Key**.
- Drag the Time slider to frame **40.**
- Press the **s** key to **Set Key**.

  You now have keys set at the following frames:

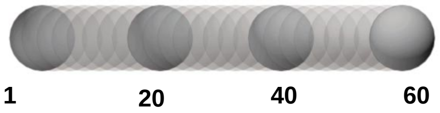


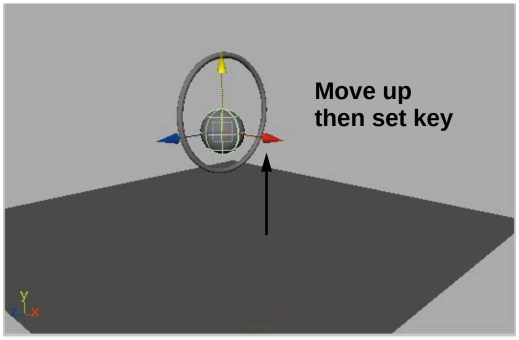
Diagram of initial keys

Tip: Every time you set a key, a red mark is placed in the Time slider to help you identify where your keys are.

5 Add more intermediate keys

To create the peaks of the bounce, you need to set new keys at frames 10, 30 and 50. These keys will require that you move the ball into a new position along the Y-axis.

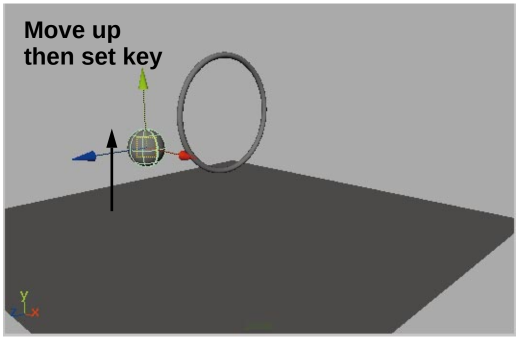
- Move the Time slider to frame 30.
- Click-drag on the green Y-axis manipulator to move the sphere up along the Y-axis until it sits inside the ring.
- Press the **s** key to **Set Key**.



Y-axis move

- Move the Time slider to frame 10.
- Click-drag on the Y-axis manipulator to move the sphere up along the Y-axis. Don’t raise it as high as you did for frame 30.

  This will put emphasis on the bounce at frame 30 where the ball is jumping through the ring.
- Press the **s** key to **Set Key**.



Y-axis move

- Repeat these steps to create the last bounce at frame 50. Don't forget to set a key after placing the ball.

6 Set the playback speed

- Click the **Animation Preferences** button found at the right side of the Time slider.

  This opens a window that lets you set various animation and playback options. Currently, the scene is playing back as fast as it can. Since you only have two objects, playback is a little too fast.

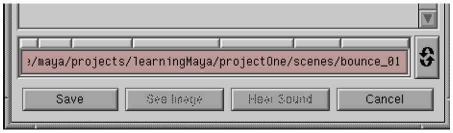
- In the **Playback** section, set the following:

  Playback Speed to **Normal.**

- Click the **Save** button.
- Playback the results using the Time slider controls.

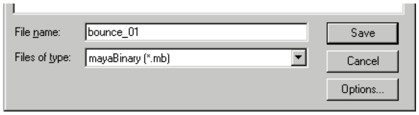
7 Save your work

- From the **File** menu, select **Save Scene As...**
- In IRIX, enter the name *bounce_01* at the end of the file's path.
- In Windows NT, enter the name *bounce_01* in the **File name box.**



:/maya/projects/learningMaya/projectOne/scenes/bounce_01

IRIX Save As dialog box



File name: bounce_01

Files of type: mayaBinary (*.mb)

Windows NT Save As dialog box

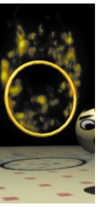
- Press the **Save** button or press the **Enter** key.

  Make sure you save this file since you will be continuing with it in the next lesson.

Note: Throughout this book, you will be using the final saved file from one lesson as the start file for the next. Save your work at the end of each lesson to make sure that you have the start file ready.

Conclusion

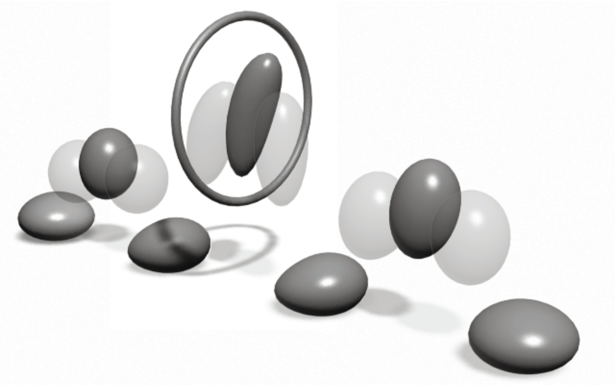
Congratulations. You have completed your first exercise using Maya. In the next lesson, you will take this simple bouncing ball and refine the motion for more dramatic results.



2 Adding Character

In the last lesson, you built and animated a bouncing ball. As you playback the bounce, you can see that the ball seems to float and is not bouncing in a convincing manner. In this lesson, you will begin to edit the quality of the original animation to add character to the bounce.

This lesson will begin to layer the animation with secondary motion such as *squash and stretch*. In Maya, you can create these effects using non-linear deformers, that allow you to reshape the ball as it animates.



Adding squash and stretch

In this lesson you will learn the following:

- How to refine animated channels with the Graph Editor
- How to create a bend and a squash deformer
- How to set keyable and non-keyable channels
- How to use Set Driven Key
- How to use Auto Key
- How to create expressions
- How to edit an animation's timing using the Dope sheet

Refining the animated channels

In the last lesson, the bouncing ball does not appear to bounce properly. It seems to float as it touches the ground instead of hitting hard as you would expect. To edit the quality of the bounce, you can use the Graph Editor to work with the animation channels.

1 Select the ball

If you are continuing straight from Lesson 1, the *ball* may already be selected. If not, then you must select this object to work with the animated channels.

- If the ball is not selected, then select it in one of the view panels.

2 Create a new panel layout

You can switch the contents of a panel to other panel types such as lists or graphs.

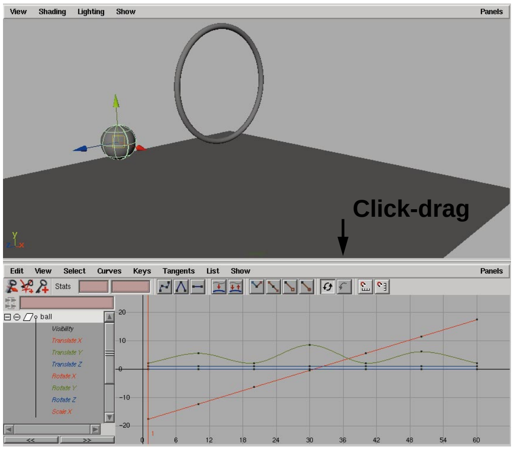
- In the perspective view panel's **Panels** menu, select **Saved Layouts → Persp/Graph**.

  This places a perspective view panel above a Graph editor panel.

- Click-drag down on the bar that separates the two panels to make the new perspective panel larger.
- From the Graph Editor's **View** menu, select **Frame All**.

  Now you can see all of the animation curves for the *ball*.

  Earlier, you set keys for the ball. This set keys for all the *Translate, Rotate* and *Scale* attributes, in all three axes, as well as *Visibility*. Each of the animated attributes creates an animation curve that represents how each channel changes over time.



Graph Editor panel

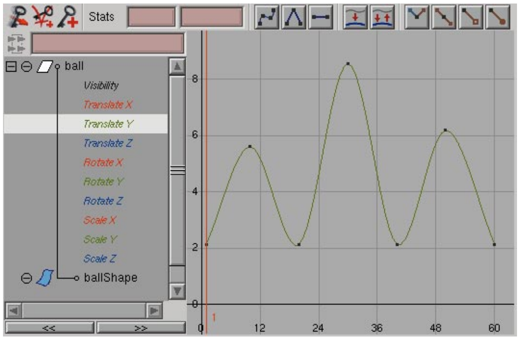
Tip: Remember that you can also access the View menu from the hotbox by pressing and holding the spacebar.

3 Focus on the Translate Y channel

To focus on the up and down motion of the ball, you need to focus on the Translate Y channel.

- Click on the *Translate Y* channel. Now only its curve is shown in the graph.
- Press the **Alt** key and click-drag with the left and middle mouse buttons to dolly into the graph.
- Press the **Shift** and **Alt** keys and click-drag up and down with the left and middle mouse buttons to dolly only along the Y-axis.
- Press the **Shift** and **Alt** keys and click-drag up and down with the middle mouse button to track along the Y-axis.

- Use these tools to position the graph as shown below:



New view of curve

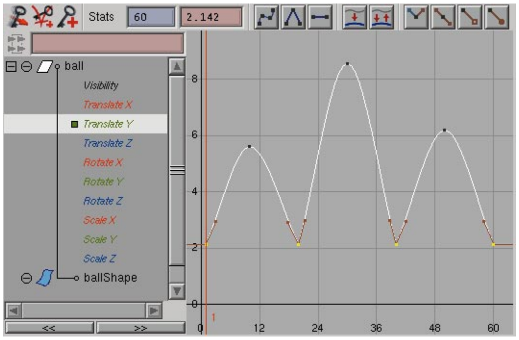
Tip: The same view tools used in the modeling views apply to other panel types. The **Shift** key constraint works in all view panels and with tumbling and tracking.

4 Edit the shape of the curve

The floating effect as the ball touches the ground is a result of the smoothing of the curve as it approaches a value of 0.

This smoothing is created by the curve tangents. The tangents define the motion in between the keys. By editing the tangents, you will change how the ball moves between the keys.

- Click-drag over all of the points at the bottom of the bounce to select the keys.
- Press the **Linear Tangents** button to make the curve tangents meet at a point.



Linear tangents

- Move the pointer over the Perspective view and press the spacebar quickly.

  This pops the panel to a full size panel. Playback is a little faster without the Graph Editor present.

Tip: You can use a quick tap on the spacebar to pop any panel to full size and back. A longer click would reveal the hotbox.

- Playback the results in the Perspective view panel.

  The resulting motion now results in more of a bounce as the ball hits the ground.

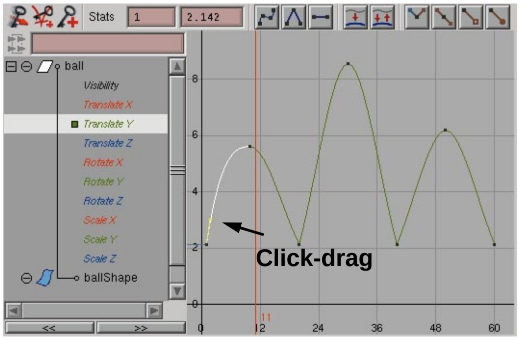
- Move your pointer over the Perspective view panel and press the spacebar quickly.

  This returns you to the two view panels you had earlier.

5 Edit the tangents at the keys

To accentuate the bounce even further, you can edit the actual tangents to make the curve steeper as it approaches the ground.

- Press the **Break Tangents** button to break the connection between the opposite tangents at the key.
- Click-drag a selection box around the tangent handle on the first key.
- Make sure that you select only the tangent handle and not the key itself.
- Make sure that the **Move** tool is active.
- Click-drag with your middle mouse button to edit the curve's tangent until it is almost vertical and you don't get a bump at the next key.

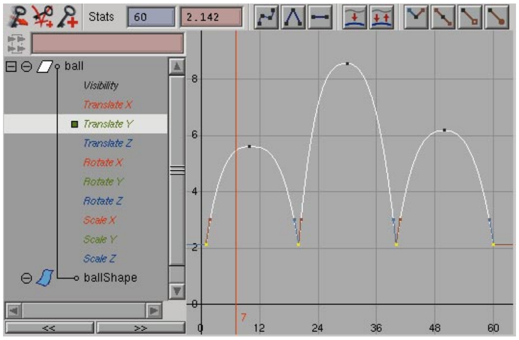


Click-drag

Edited tangent handle

- Click-drag a selection box around the next key on the curve.

  This reveals the tangent handles for that key.
- Click-drag a selection box around one of the tangent handles.
- Edit the tangent using the middle mouse button to make it more vertical.
- Repeat the tangent edit for all of the other tangents until the curves appear as shown:



Action curve with keys highlighted

- Playback the results in the Perspective view.

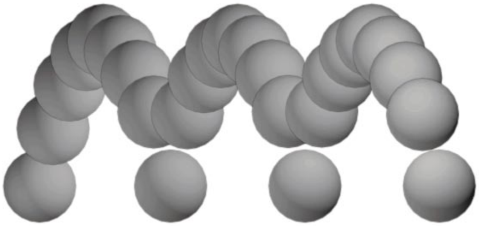


Diagram of edited bounce

The bounce is much more sharp and appears to be affected by gravity.

Cleaning up your curves

Now that the quality of the motion has been set, you can look for ways to clean up your animation curves so that unnecessary information is removed. For this scene, the extra curves will have little influence, but in larger scenes they can add up and affect performance.

1 Remove static channels

If an action curve is flat along its whole length then its value is not changing. This means that it is not being animated.

These curves should be removed to reduce the amount of data in the scene.

- In the name list, press the **Ctrl** key and click on the following channel names:

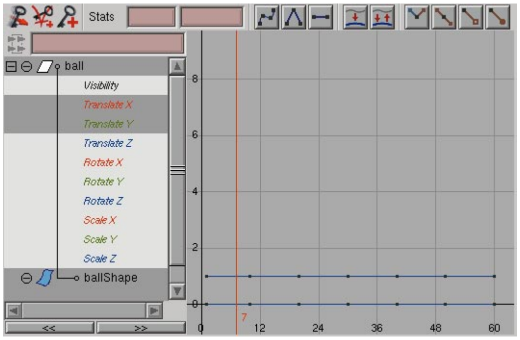
  Translate Z;

  Rotate X, Y and Z;

  Scale X, Y and Z;

  Visibility.

  The *Translate X* and *Translate Y* should remain unselected.



Static channels

- From the Graph Editor's **Edit** menu, select **Delete**.

  This removes the selected curves from your scene.

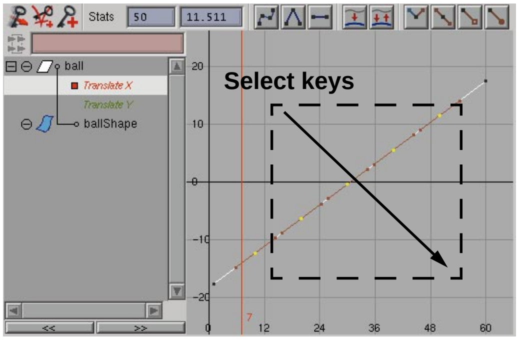
Note: In the Channel box, you will see that the remaining input fields – *Translate X* and *Translate Y*– use a different background color (green on IRIX, orange on Windows NT). This color helps you distinguish channels which have been keyed.

2 Remove extra keys

In some cases, you may have keys on animated channels that don't contribute to the results. These keys are found when a straight line can be drawn from the key before to the key after.

- In the name list, click on the *Translate X* channel.
- Select **View** → **Frame All**.
- Use the **Alt** key to track and dolly so you can see the whole curve.

  You will see that there are five redundant keys on this curve. Since you want the ball to move along a straight path, these keys are not required.
- Click-drag a selection box around the five middle keys.

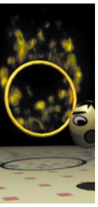


Extra keys

- Press the **Delete** key to remove the keys.

  The resulting curve would now be easier to edit because there are no intermediate keys.

SQUASH AND STRETCH



One of the best methods to infuse your animation with life is to add *squash and stretch.* This traditional animation technique helps you avoid a rigid looking animation as the ball hits the ground with more emphasis and seems to soar as it leaps into the air. In Maya, you can create this effect using a squash deformer.

Adding a squash deformer

A deformer is a special object type that lets you reshape a surface or group of surfaces. By using deformers to change the shape of the object over time, you can create a more visually appealing animation.

1 Set up a front view

- Go to frame 1.

  This will place the ball at its starting point.

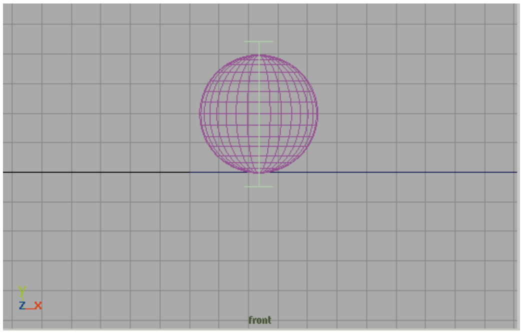
- In the perspective view's **Panels** menu, select **Panels** → **Orthographic** → **Front**.

  This gives you a view of the ball looking down the Z-axis.

- Dolly and track the view until you are looking at the side of the ball.

2 Add the squash to the ball

- Select the *ball.*
- Press **F2** to go to the **Animation** menu set.
- From the **Deform** menu, select **Create Nonlinear** → **Squash**.
- This adds the *squashHandle* to the ball. You do not see any deformation yet since the squash attributes have not been edited.



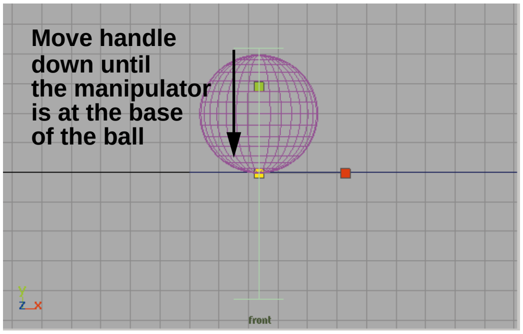
Scale deformer

3 Move and Scale the deformer

- **Move** the *squashHandle* so that the manipulator is at the base of the ball.

- **Scale** the squashHandle until the handle appears at the top of the ball.

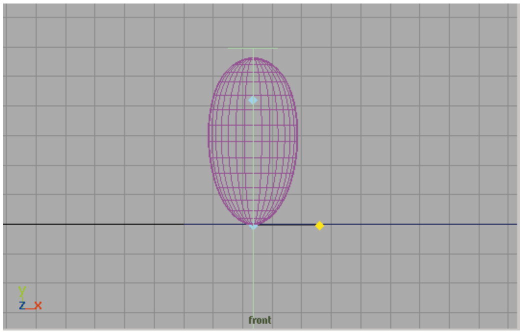
  Now the squash will occur from the base of the ball instead of in its middle. This ensures that the bottom of the ball doesn't leave the ground during the squash.


Move handle down until the manipulator is at the base of the ball
Y
Z X
front

Scale deformer

4 Test the squash's Factor attribute

- In the Channel box, click on the *squash* input node.
- Select the **Show Manipulator** tool.
- Click-drag on the middle manipulator handle to squash and stretch the ball.
- When you are finished check the channel box and make sure that the **Factor** attribute is set to about **0.4** to give the ball a little stretch.

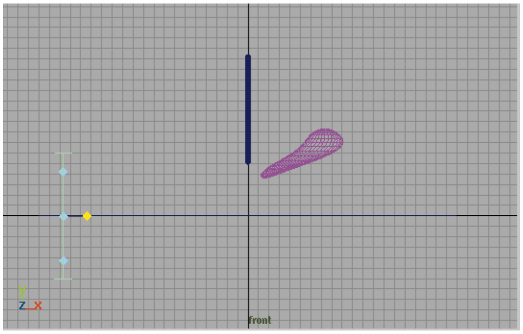


Scale deformer

5 Test the playback of the ball

- Dolly out of the view so that you can watch the whole bouncing motion.
- Playback the animation.

  The ball animates but the deformation of the ball only works properly at frame 1. This is because the deformer is not moving with the ball.



Playing back the animation

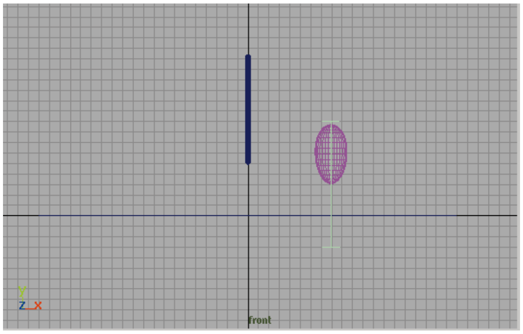
6 Parent the deformer

To make the deformer move with the ball, you need to parent the deform handle to the ball.

- Go back to frame 1.
- Select the *squashHandle* then press the shift key and select the *ball*.
- Select **Edit** → **Parent**.

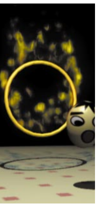
  You could also press the **p** key to parent the handle to the ball.
- Playback the animation.

  Now the deformer moves with the ball.



Animating the parented nodes

Animate the squash



To animate the ball's squash and stretch, you could set keys, but if you later edited the timing of the ball's bouncing then these keys would have to be reset. For this reason, you will use a *reactive* animation technique called **Set Driven Key**.

This technique allows you to have one attribute drive another attribute. Since you know that you want the ball to squash when it touches the ground and stretch when it is in the air, you can use the ball's *Y translation* to drive the squash's *factor* attribute. Therefore if you later edit the timing of the ball's bouncing or change its *Y translation*, the squash will *react* appropriately and squash on cue.

1 Load the Set Driven Key window

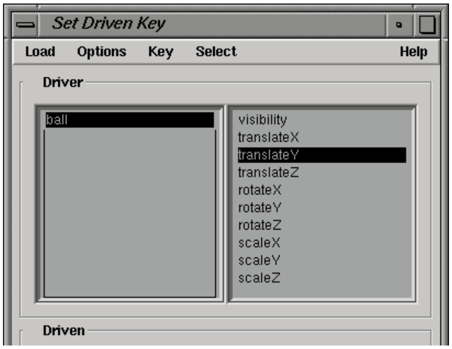
The Set driven key window lets you choose which attribute will be the driver and which attribute(s) will be driven.

- From the **Animate** menu, select **Set Driven Key → Set - ❐** to open the Set Driven Key window.

Note: When you want to open an option window, you need to select the menu item then move your cursor over the square icon – ❐.

- Select the *ball.*
- Click on the **Load Driver** button in the Set Driven Key window.
- In the right hand column of the **Driver** section, click on **translate Y**.

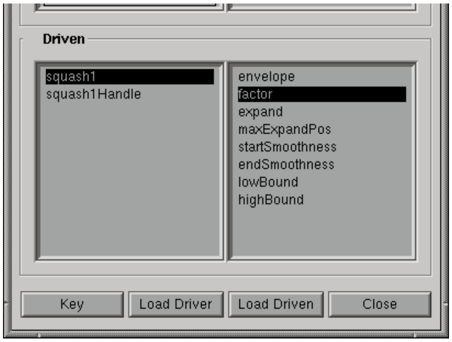
  This sets up *translate Y* as the attribute that will drive the action.



Driver attribute highlighted

- **Select** the *squashhandle*.
- Highlight the **squash** input node in the Channel box.
- Click on the **Load Driven** button in the Set Driven Key window.

  Both the *squash* input node and the *squashHandle* node are loaded. You will only be using the *squash* node.
- In the left hand column of the **Driven** section, click on **squash**.
- Click on **factor** in the right hand column to highlight it as the driven attribute.



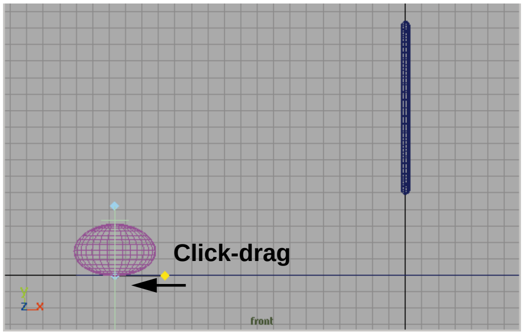
Driven attribute highlighted

2 Key the initial position

- Go to frame 1.
- Again highlight the **squash** input node in the Channel box.
- Set the *squash* node's **factor** attribute to show the ball in a squashed position.

  The image below shows a value of about -0.5.

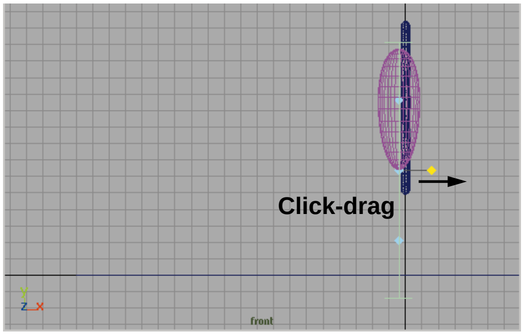
  You can either use the Channel box or the **Show Manipulator** tool.
- Click on the **Key** button in the Set Driven Key window.



The squashed ball

3 Key the second position

- Go to frame 30.
- Set the *squash* node's **factor** attribute to show the ball in a stretched position.
- Click on the **Key** button in the Set Driven Key window.



The stretched ball

- Playback the animation

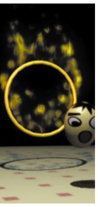
  Now the ball stretches at the peak of each bounce and squashes as the ball hits the ground. Set Driven key is using the height of the ball to decide whether to use squash or stretch.

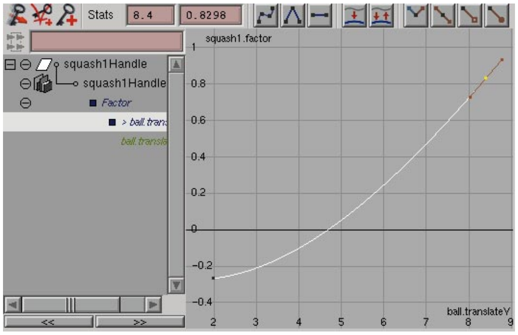
4 Edit the Animation curve

- Select the *squashHandle*.
- In the Graph editor, click on the attribute labelled > *ball.translate Y*.
- From the editor's **View** menu, select **Frame all**.

  Now you can see a straight line mapping the ball's translate Y attribute to the squash's factor attribute.
- Edit the curve as shown in the following image.

  Now the ball will have a more subtle squash when it is near the ground and an exaggerated stretch when the it is high in the air.





Edited animation curve

- Playback the animation

  You can see that there is a little more emphasis on the point where the ball hits the ground.

Bending the ball

As the ball goes into and out of a bounce, it might be helpful to have it lean forward as it bounces up and lean back as it falls. You can do this with a second deformer. This time you will use a bend deformer.

This will help give the appearance that the ball is propelling itself forward each time it hits the ground. This will give the ball a life of its own as it bounces along.

1 Add a Bend deformer

- Go to frame 1.
- Select the *ball*.
- From the **Deform** menu, select **Create Nonlinear → Bend**.

  Just like the Squash deformer, the Bend adds a *bendHandle* and a *bend* node to the ball. The *bendHandle* node is currently highlighted.

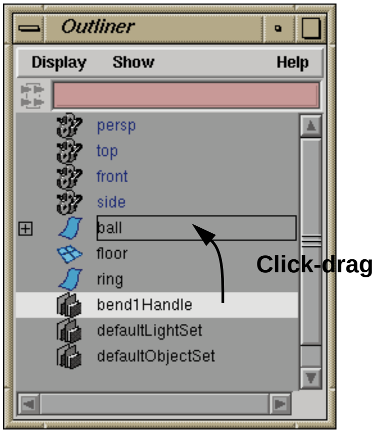
2 Parent the deformer

To make the deformer move with the ball, you need to parent the deform handle to the ball.

- From the **Window** menu, select **Outliner**.

  This window offers you a listing of all the objects in the scene.

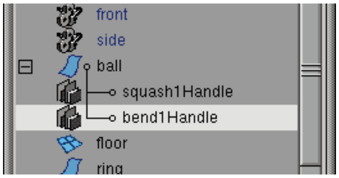
  With your middle mouse button drag the *bendHandle* onto the ball node.



Parenting the bendHandle

- Click on the plus sign next to the ball node to open up the hierarchy.

  Now you can see that the ball is the parent of the two deformer handles.



Viewing the hierarchy

- Close the outliner window.

3 Key the bend node's curvature

The bend node has an attribute called curvature that is used to control how much bending will occur. This is the attribute that you will set keys on in order to animate the bending of the ball.

- In the Channel box, click on the *bend* Input node to open it up.
- Click on the **Curvature** attribute's name to highlight it.
- From the Channel box's **Channels** menu, select **Key Selected**.

  This sets a start key for the curvature attribute.

4 Turn on Auto Keying

To set more keys for the curvature, you will use Auto key. This method of keying lets you set keys by simply changing the attribute value.

- Turn on Auto Key.

  Now when you update an attribute that has existing keys, a key will be set automatically.

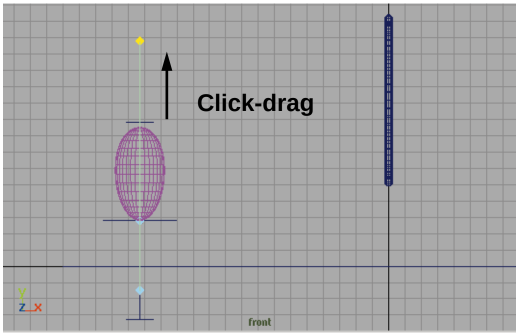
5 Edit the bend's boundary

- Go to frame **5**.
- Make sure that the **bend** input node's name is highlighted in the channel box.
- Select the **Show Manipulator** tool.

  This tool displays some handles that let you edit the attributes of the bend deformer. This gives you interactive control over the deformer's attributes.

- Click drag on the top manipulator to extend the **high bound** of the deformer.

  This will ensure that the bending works properly when the ball is fully stretched.

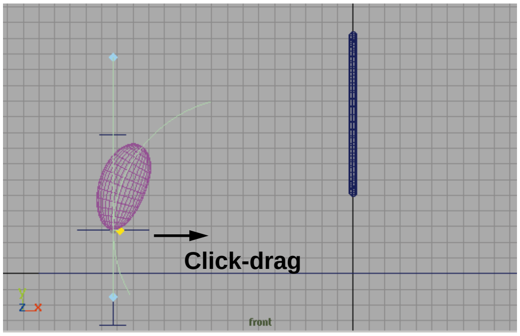


Updated high bound

6 Edit the curvature

- Click-drag to the right on the center manipulator handle to edit the bend node's **curvature**. Stop when the ball is bending forward. The value in the Channel box should read about 0.5.

  A new key has been set. Auto Key creates the new key when you adjust the curvature attribute.

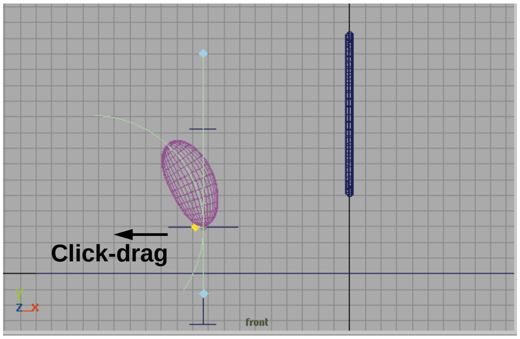


Bending forward

7 Set another key

- Go to frame **15**.
- Drag to the left on the center manipulator handle to bend the ball back. This will set the node's **curvature** to about -0.6.

  Another key is set by Auto Key.



Shear backward

8 Continue animating the bending

- Use the Auto key method to set new keys for the bend node's **curvature** so that the ball bends back at frames 25 and 45 and forward at frames 35 and 55.
- Go to frame 60 and change your **curvature** to **0**.
- Change the front view panel back to a perspective panel.
- Playback the results.

Editing timing

To give the impression that the ball is pushing off as it bounces, you will have to edit the timing of the bend action. The goal is to move the bend action closer to the bounce of the ball to make it appear more immediate. You can edit the timing of your keys in the Graph Editor.

Earlier, you edited the quality of the animation using the Graph Editor. To edit the timing of the bouncing, you can also use the *Dope sheet*. The Dope sheet lets you focus on timing without having to look at all of the curve information.

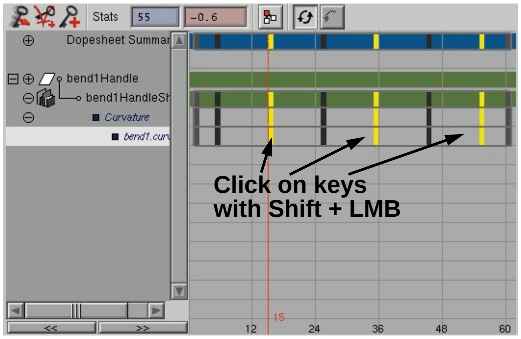
1 Select keys in the Dope sheet

- Make sure the *bendHandle* node is selected and the *bend* node is not highlighted in the Channel box.
- In the Graph editor panel's **Panels** menu, select **Panel** → **Dope Sheet**.

  This window shows each channel as a series of ticks on a long bar which represent frames. Some of these are highlighted with color bars that show you where keys have been set. The top bar shows a summary of the keys while each channel of the selected nodes has its own bar.
- Click on the plus sign next to the *bendHandle* node that is in the square.

- Click on the second plus sign next to the *bendHandleShape* node that is in the circle.
- With the left mouse button, click on the key at frame **15**.

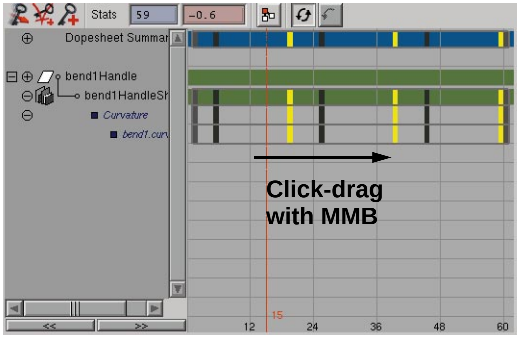
  It will be the only key highlighted in yellow.
- Press the **Shift** key and click with the left mouse button on the keys at frames **35** and **55**.



Selected keys in Dope sheet

2 Move the ballShear keys

- Select the **Move** tool.
- With the middle mouse button, click-drag to the right in the Dope sheet until the first key is at about frame 19.



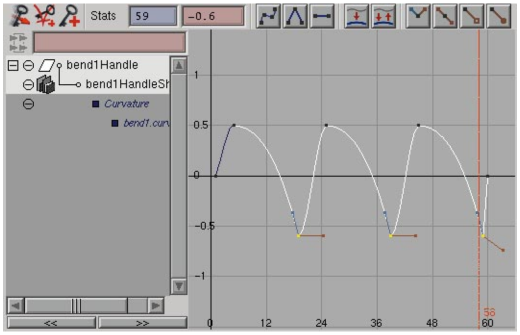
Edited keys

- Playback the results.

3 Edit the shape of the curve

While editing the timing of the bending in the dope sheet helps a bit, you may not be getting the look that you want. To edit how the curve reacts between the keys, you can edit the shape of the curve in the Graph editor.

- In the Dope Sheet's **Panels** menu, select **Panel** → **Graph Editor**
- Use the techniques learned earlier in the lesson to reshape the curve as shown below:



Reshaped curve

The key is that when the ball hits the ground, the bending changes sharply to emphasize the bounce.

- Playback the results.

4 Save your work

- From the **File** menu, select **Save Scene As...**
- Enter the name *bounce_2* next to the file's path.
- Press the **Save** button or press the **Enter** key.

Tip: It is a good idea to give your file a new name after significant changes have been made. The different files give you a record of past stages in case you want to go back. Once you have saved a few versions, you can clear earlier files to help conserve disk space.

Conclusion

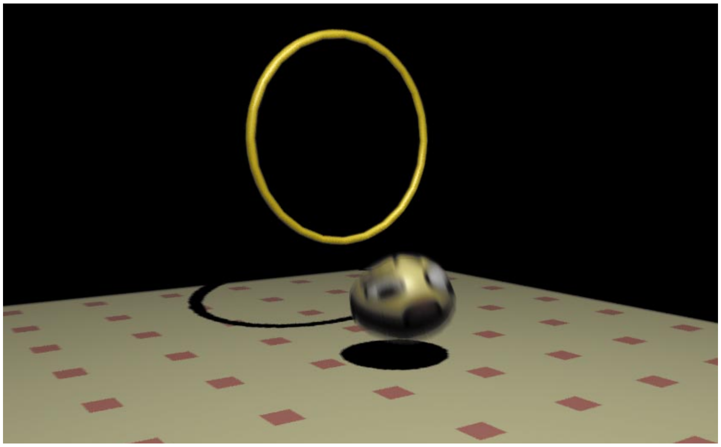
You have now added some character to the bouncing ball. In the next lesson, you will learn how to texture and light your scene in order to begin creating rendered animations.

If desired, you could now use the Dope sheet or the Graph editor to further refine the quality and timing of the various channels to achieve a bounce with different kinds of character. You might want the ball to bounce higher or the bending to be more subtle.

This refinement is ultimately your role as an animator. While the actual work of setting keys is finished, you will find that the refinement stage could become the most time-consuming, and yet the most rewarding. Since this book is focused on technique, the refinement will be left up to you to add at the end of each lesson.

3 Rendering

Now that you have animated the ball, you are ready to render the scene. The rendering process involves the preparation of shaders and textures for your objects and the manipulation of cameras and lights. You can also add effects such as *motion blur* to accentuate the motion during playback of the rendered frames.



Software rendering of the ball

In this lesson you will learn the following:

- How to work with a menu-less user interface
- How to work with the Hypershade panel
- How to create a shading group
- How to texture map an object
- How to add lighting to your scene
- How to test render a single frame
- How to set up motion blur
- How to software render an animation

Hiding the general UI

In the last two lessons, you used menus, numeric input fields and other user interface elements to work with your scene. In this lesson, you will hide most of the user interface and rely more on the hotbox and other hotkeys that let you access the UI without actually seeing it on screen.

1 Turn off all of the menus

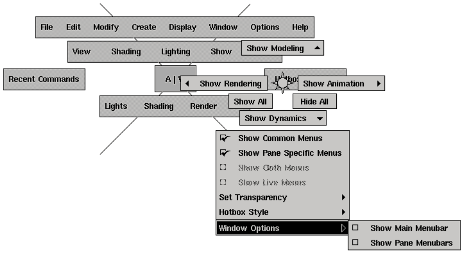
- Move your cursor over the Perspective view panel, then press the spacebar quickly to pop this panel to full screen.
- Press and hold on the spacebar to evoke the hotbox.

Tip: Tapping and holding down the spacebar, respectively, can be used to both toggle between window panes and to bring up the hotbox.

- Click on the **Hotbox Controls**.
- From the marking menu, go down to **Window Options** and set the following:

  Show Main Menubar to **Off**;

  Show Pane Menubars to **Off**.

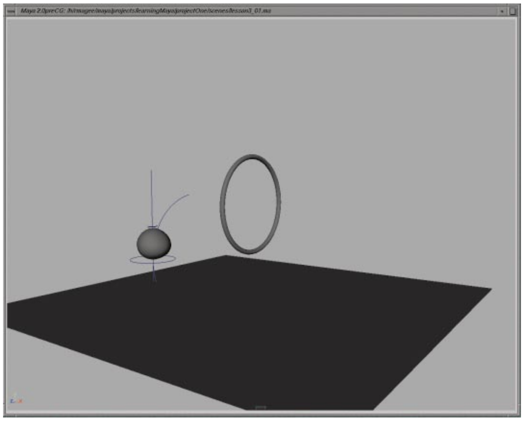


Marking menu

Now the various menus are hidden and you must rely on the hotbox to access tools.

2 Turn off all of the workspace options

- From the hotbox, select **Options** → **Show Only Viewing Panes**.

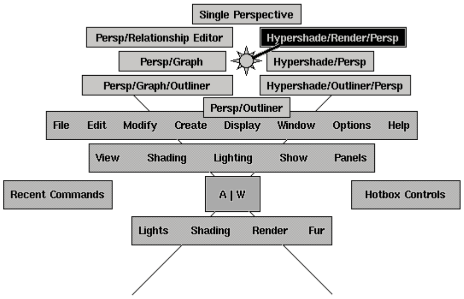


Simplified user interface

You now have a much larger working area which will let you focus more on your work.

3 Change the panel organization

- Press and hold on the spacebar to evoke the hotbox.
- Click in the area above the menus to invoke a marking menu.
- Select **Hypershade/Render/Persp** from this marking menu.


Single Perspective
Persp/Relationship Editor
Hypershade/Render/Persp
Persp/Graph
Hypershade/Persp
Persp/Graph/Outliner
Hypershade/Outliner/Persp
Persp/Outliner
File Edit Modify Create Display Window Options Help
View Shading Lighting Show Panels
Recent Commands
A | W
Hotbox Controls
Lights Shading Render Fur

Marking menu

Tip: Each of the four quadrants surrounding the hotbox and the hotbox's center all contain their own marking menu set. You can edit the contents of these menus using **Options → Customize UI → Marking Menus**...

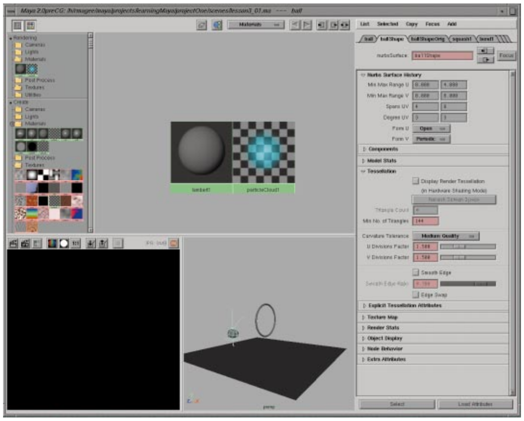
This saved layout puts a single Perspective panel above a Hypershade and a Render view panel.

The Hypershade is where you will build shading networks and the Render view is where you will test the results in your scene.

4 Open the attribute editor

- From the **Options** menu, select **Attribute Editor**.

  Now you also have an attribute editor panel on the right side of the workspace. This will make it easy to update shading group attributes.



New window layout

Hotkeys

When working with a minimal UI, you will rely on the hotbox and hotkeys for your work. The following is a list of relevant hotkeys that you may need to use as you work:

spacebar	hotbox/window popping
Alt v	Start/stop playback
Alt + Shift v	Go to first frame
Alt .	Move one frame forward
Alt ,	Move one frame back
k	Click-drag to scrub animation

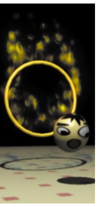
If you have selected an animated node such as the ball, then you can also use the following hotkeys:

.	Go to next key
,	Go to last key

For a complete listing of available hotkeys, go to the **Options** window and select **Customize UI → Hotkeys...**

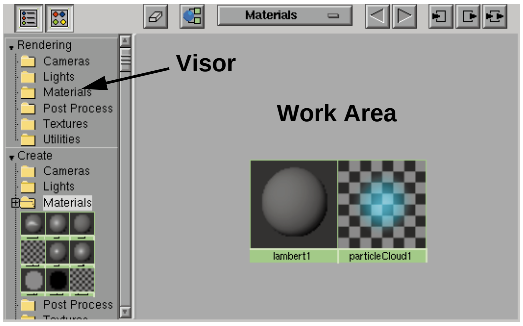
SHADING GROUPS

To prepare the ball and the floor for rendering, you need to add color and texture. In Maya, this is accomplished using *shading groups* that bring together material qualities, textures, lights and geometry to define the desired look.



The Hypershade panel

The Hypershade panel is made up of two sections – the work area and the Visor. The Visor gives you access to nodes that can be dragged to the work area.



Closeup of Hypershade

Creating a shading group

A shading group is made up of a series of nodes which input into a *shading group and material nodes*. In the following examples, you will create several nodes that define the material qualities of the ring and the floor.

1 Create a material for the ring

To build a material for the ring, you will use the Hypershade and the Attribute editor.

- Click with your right mouse button in the Hypershade panel and select **Graph → Clear View.**

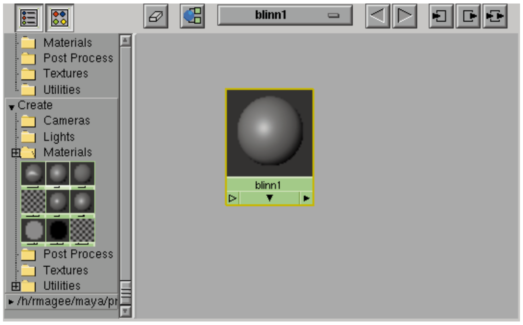
This clears the workspace so that you can begin working on a new shading network.

- In the Hypershade's Visor section, scroll down to the **Create** section then click on the **Materials** folder.

This offers you a series of icons that can be used to create new materials.

- Move your cursor over the icons until you find one that says *blinn# (blinn)*.
- With your middle mouse button, click on the **Blinn** icon and drag it into the working area of the Hypershade.

This adds a new blinn material to your scene file that can be assigned to objects. You will also see the Attribute editor update to show the new nodes information.



New node in Hypershade

Blinn is a particular type of shading model that defines how the material will look. The Blinn model gives you control over the look of the materials highlights using special attributes.

2 Rename the material node

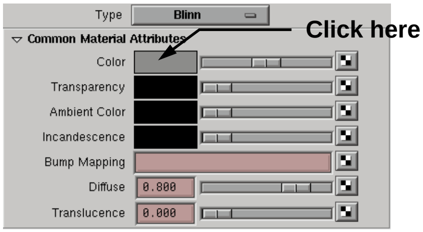
- Press **Ctrl - a** to open the Attribute editor. In the Attribute editor, change the name of the material node to *ringM*.

  The *M* designation is to remind you that this node is a material node.

3 Edit the Material's color

To define how the material will render, you will need to set several key material attributes such as color.

- In the Attribute editor, click on the color swatch next to the **Color** attribute.

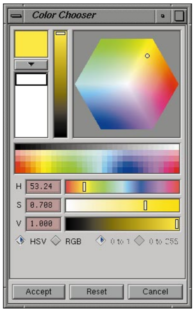


Type Blinn
Common Material Attributes
Color
Transparency
Ambient Color
Incandescence
Bump Mapping
Diffuse 0.800
Translucence 0.000

Click here

Color swatch in the Attribute Editor

This opens the Color Chooser. This window lets you set color by clicking in a color wheel and editing HSV (Hue, Saturation, Value) or RGB (Red, Green, Blue) values.



Color Chooser
H 53.24
S 0.708
V 1.000
◆ HSV ◇ RGB ◆ 0 to 1 ◇ 0 to 255

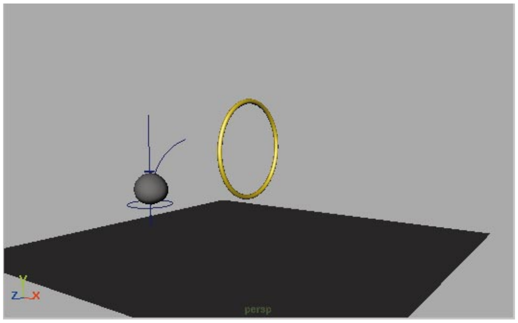
Color Choosers

- Choose any color you want then click the **Accept** button.

4 Assign the material

- With your middle mouse button, click-drag on the *ringM* node and drag it from the Hypershade panel into the perspective view and drop it on the ring.

  This assigns the material to the object.



Assigned shader

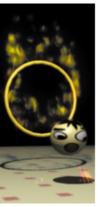
Creating a texture map

To give the floor a pattern, a grid texture will be added to the *floor* material's color. This will be accomplished using the drag and drop capabilities of the Hypershade.

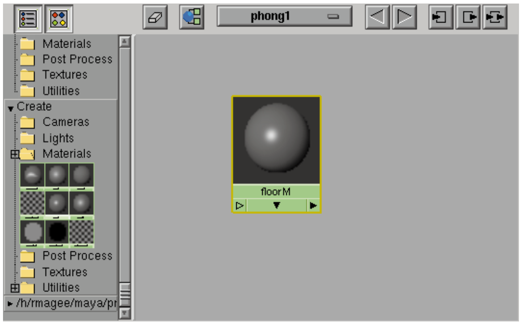
1 Create a material for the floor

To build a textured material for the floor, you will use the Hypershade panel to build up the shader using a grid texture.

- From the Hypershade panel's **Graph** menu, select **Clear View**.
- In the Hypershade's Visor section, move your cursor over the icons until you find one that says *phong# (phong)*.



- Click with your middle mouse button on the **Phong** icon and drag it into the working area of the Hypershade.
- In the Attribute editor, change the name of the material node to *floorM*.



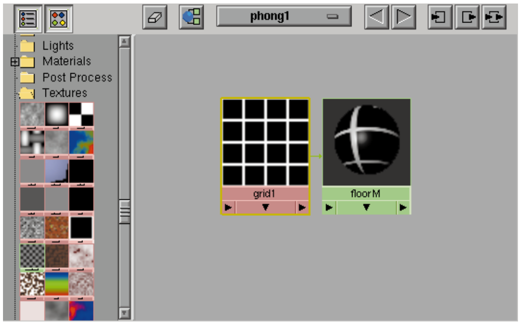
New node in Hypershade

2 Create a checker texture

- In the Hypershade's Visor section, scroll down to the **Create** section then click on the **Textures** folder.

  This offers you a series of icons that can be used to create new textures.
- Move your cursor over the icons until you find one that says *grid# (grid)*. It's icon appears all black because of its reduced size.
- Click with your middle mouse button on the **Grid** icon and drag it onto the **floorM** material node in the working area of the Hypershade.
- Release the mouse button.

  A pop-up menu appears offering you a number of attributes that can be mapped by the checker texture.
- Select **color** from the menu to map the grid to the material node's color attribute.

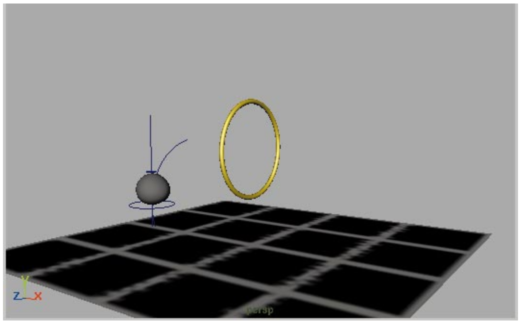


Texture node

3 Assigning the material

You will assign the texture map to the floor then use hardware shading.

- With your middle mouse button, click on the *floorM* material node and drag it onto the floor surface in the perspective view.
- Move your cursor over the Perspective window and click with your middle mouse button.
- From the hotbox, select **Shading** → **Hardware Texturing.**



Hardware texturing

Tip: You can also turn on hardware texturing by making the desired panel active and pressing the **6** key.

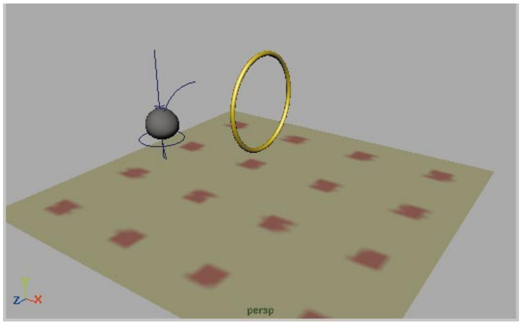
4 Edit the grid attributes

- In Hypershade, click on the *grid* node.
- In the Attribute editor, click on the color swatch next to the **Line Color** attribute.
- Choose any color you want then click the **Accept** button.
- Click on the color swatch next to the **Filler Color** attribute.
- Choose any color you want then click the **Accept** button.
- Change the grid's width attributes as shown below:

  U Width to **0.75**;

  V Width to **0.75**.

The Attribute editor allows you to easily update the look of a procedural texture.



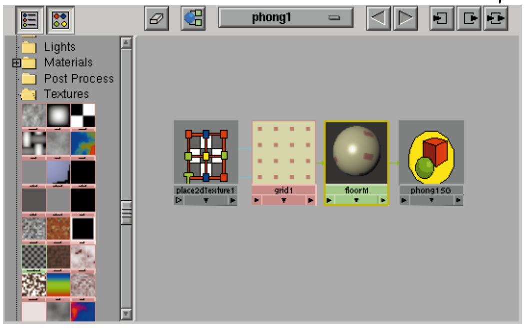
New Grid

5 Display the whole shading group

- With the *grid* texture highlighted in the Hypershade, go to the upper right of the window and click on the **Up and Downstream Connections** button.

  This displays some other nodes that help define this shading group.
- Press the **Alt** key then click-drag with your left and middle mouse button to zoom out.

Up and Downstream Connections



Complete shading network

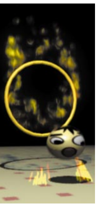
6 Edit the texture's positioning

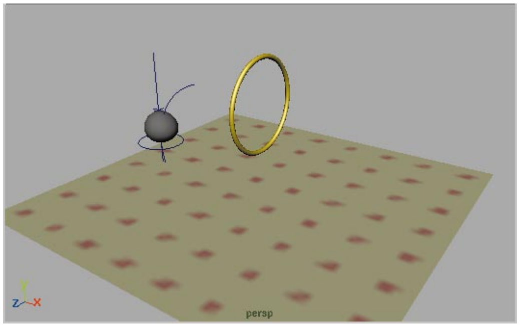
The placement of a texture on a surface is defined by the *place2DTexture* node.

- Click on the *place2dTexture* node to highlight it.
- In the Attribute editor, change the following attributes:

  Repeat U to **8**;

  Repeat V to **8**.

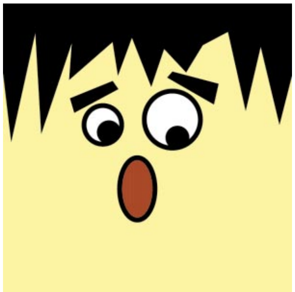




Updated texture placement

Creating a ball material

You will create a material for the ball that uses a file texture instead of a procedural texture. Many digital artists like to create textures in a 2D paint package. For the ball, you will use a painted texture of a face.



Face texture

1 Create a material for the ball

- From the Hypershade panel's **Graph** menu, select **Clear View**.
- Click-drag another **Phong** material node into the working area of the Hypershade.
- Rename this node *ballM.*

2 Create a file texture node

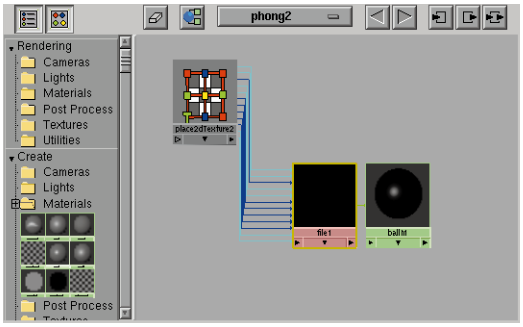
To load an external texture, you need to start with a file texture node.

- In the Attribute editor, click on the **Map** button next to **Color**. The map button is shown with a small checker icon.

  This opens the Create Render node window.

- Click on the **Textures** tab in the **2D Textures** section, then click on **File**.

  A file node is added to the phong material. All the appropriate connections have already been made.



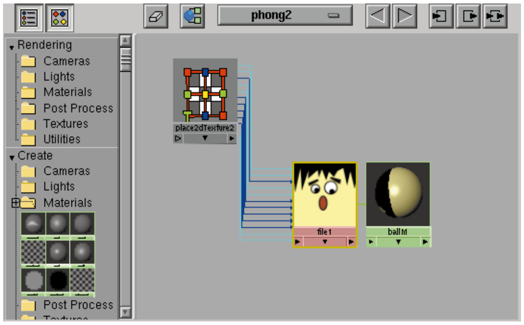
New file texture node

- Close the Create Render Node window.

3 Import the file texture

- In the Attribute editor, click on the **File folder** button next to **Image name**.
- Select the file names *face.iff* then click on the **Open** button.

  The file texture is now loaded into the shading network.



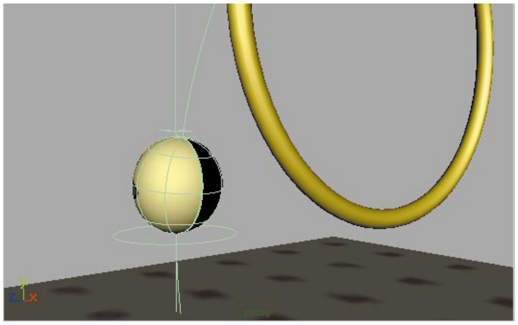
Imported file texture

Note: This file will be available only if you set up your project as indicated at the beginning of lesson 1.

4 Apply the material to the ball

- **Select** the ball in the perspective view.
- In the Hypershade, click on the new material node with your right mouse button and choose **Assign Material to Selection** from the pop-up menu.

  The texture is assigned to the ball but it doesn't seem to be positioned correctly. You will correct this using the placement node.



Ball with texture

Tip: This method of assigning materials works better than the click-drag method when you want to assign a material to multiple objects.

Positioning the texture

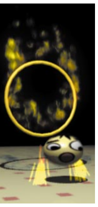
When the file texture was assigned to the ball, it was mapped to the actual surface of the ball based on the surface's topology. In this case, this results in the texture being rotated on the surface of the ball. You can rotate the texture's placement to correct the problem.

1 Change the texture's rotation

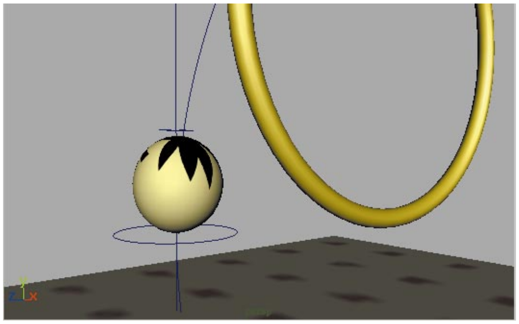
- Click on the file texture node's *place2DTexture*. This will activate it in the Attribute Editor.
- In the Attribute Editor, set the following attribute:

  Rotate UV to **90**.

  If you are using hardware texturing, you can see the file texture is now oriented properly on the ball's surface but it is facing the wrong



way. This is because the ball's seam is facing forward. You need to offset the texture.

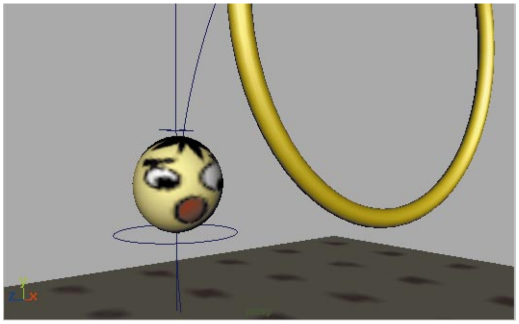


Backward ball texture

- In the Attribute Editor, set the following attribute:

  Offset V to **0.5**.

  This will place the file texture so that it is facing the front of the ball.



Rotated ball texture

2 Test Render the scene

Now that you have the textures ready, it would be a good time to do a test rendering.

- In the Render view panel, click with your right mouse button and select **Render** → **Render** → **persp** from the pop-up menu.

  Oops! The scene renders to black.

  This is because there are no lights in the scene. You must first add a few lights to illuminate the objects for the renderer.

LIGHTING

In the real world, it is light which allows us to see the surfaces and objects around us. In computer graphics, digital lights play the same role. They help define the space within a scene, and in many cases help to set the mood or atmosphere.

Placing a spot light

To create the primary light source in the scene, you will use a spot light. This light type lets you define important attributes such as the light's cone angle and its intensity.

1 Create a spot light

- Press **F5** to go to the rendering menu set.
- Click on the perspective view panel with your middle mouse button then press the **spacebar** quickly to pop to a full view.
- From the hotbox, select **Lights** → **Create Spot Light.**

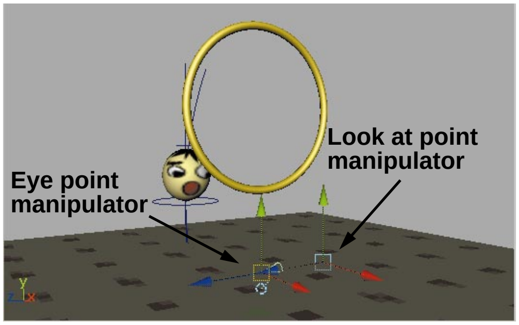
  This places a light at the origin.

Note: In the hotbox, you will find a lights menu and a lighting menu. The lights menu is for adding lights to the scene while the lighting menu lets you control the panel's hardware lighting.

2 Edit the spot light's position

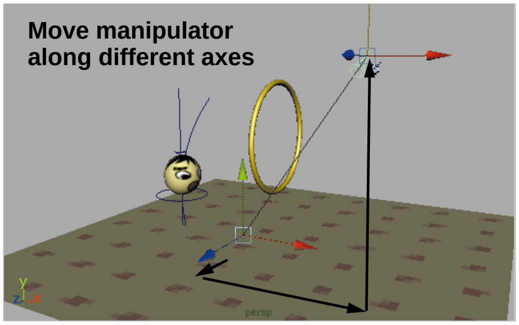
This tool gives you a manipulator for the light's *look at point* and *eye point*. You can edit these using the same method as you would with a typical transform manipulator.

- Press the **t** key to access the **Show Manipulator** tool.



Show manip tool

- Click-drag on the manipulator handles to reposition the light.
- Move the manipulators until they appear as shown below.



New light position

3 Turn on hardware lighting (if possible)

One step beyond hardware texturing is *hardware lighting*. This lets you see how the light is affecting the surface that it is shining on.

- From the hotbox, select **Lighting → Use All Lights.** This time you will use the other Lighting menu. You could also use the **7** key.

  At first you may not see much lighting on the surface. You need to update the floor's subdivisions.

- In the Perspective window, click and hold with the right mouse button on the floor object.
- From the marking menu, select **Inputs → PolyPlane - ❏**.

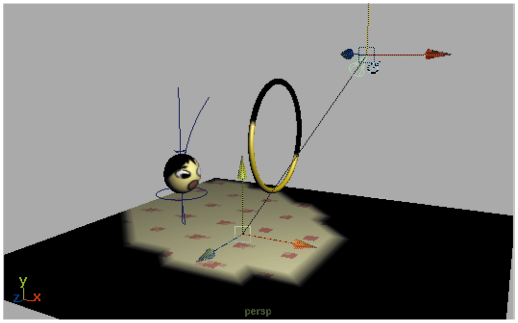
  This node now displays in the Attribute editor.

- Set the following attributes:

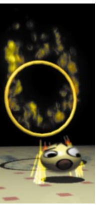
  Subdivisions X to **20**;

  Subdivisions Y to **20**.

  The spot light's cone of illumination can now be seen on the surface of the floor.



Increased subdivisions on floor surface



Note: You only need to increase the subdivisions for a hardware texturing preview of the scene. If you are software rendering, then one subdivision in each direction should work fine.

4 Edit the spot light's cone angle

You can now edit some of the light's attributes to control its effect. You will reveal other light manipulators to let you edit this attribute interactively.

- **Select** the spotlight.
- Press **t** to select the **Show Manipulator** tool.

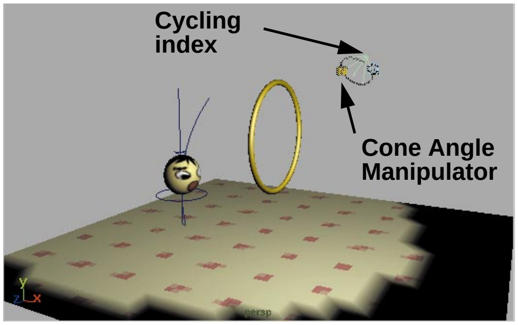
Next to the light is a small icon that displays a circle with a small line pointing up and to the right. This icon is the cycling index and is used to cycle between different types of light manipulators.

- Click two times on the manipulator's cycle index.

  The cycling index rotates to show that you are accessing new manipulators. The chosen manipulator consists of a little blue dot just outside of the light cone. The new manipulator lets you edit the cone angle of the spot light.

- Click-drag on the cone angle manipulator to illuminate more of the scene.

  In the Attribute editor, you can watch the **Cone Angle** attribute update as you click-drag.



New angle of view

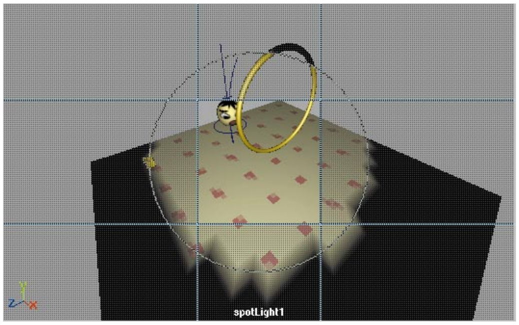
5 View through the light

It is sometimes difficult to position a light when standing beside the scene. In Maya, you can also look *through* the light to aid in positioning.

- Move your cursor over the perspective view panel.
- Using the hotbox, select **Panels** → **Look Through Selected**.

  You are now looking at the scene from the light's point of view. The Cone Angle manipulator is still available for you to click-drag.

- Click-drag on the manipulator to set the **Cone Angle** to about **60** degrees.



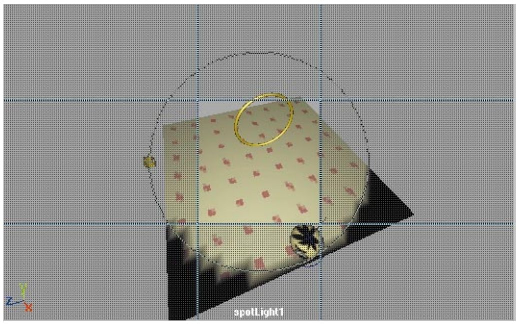
Spot light view

6 Reposition the light

You can now use the view change tools to edit the position of the light.

- Press the **Alt** key to tumble, pan and dolly into the view.

  Work with these tools until you see a view similar to the view shown below:



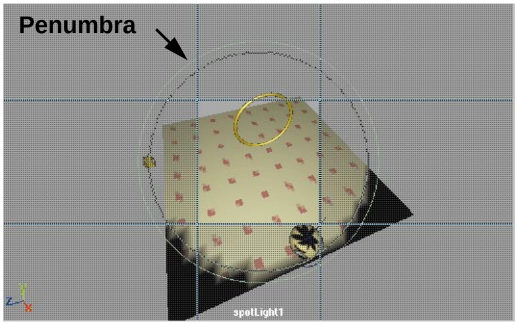
New spot light view

Tip: You may want to playback the animation to make sure that the ball is illuminated throughout its bouncing.

7 Adjust the Penumbra angle

To get some softness at the edge of the spotlight, you can adjust the light's penumbra.

- In the Attribute editor, set the **Penumbra Angle** to **5**.
- You can now see a second circular line outside the cone angle icon that indicates the area where the light will be soft.



New spot light view

8 Return to Perspective view

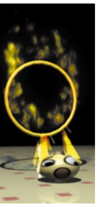
- Move your cursor over the perspective view panel.
- Using the hotbox, select **Panels** → **Perspective** → **persp.**

You are now looking at the scene from your old point of view.

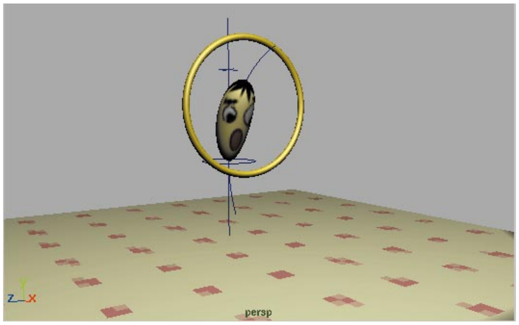
Tip: You can click-drag lights and cameras from the Outliner to any view panel.

9 Playback the animation

- Use the view tools to center the action within the view panel.



- Playback the animation using the necessary hotkeys to see how it looks with hardware lighting.



Perspective view

Rendering the scene

Now that you have a light in your scene, you can turn on shadow casting then create a test render. This will let you see a higher quality image of a still frame.

1 Turn on depth map shadows

In Maya, shadows are turned off by default for each light. This allows you to turn shadows on for only those lights that require shadows.

- Select the *spotLight*.
- In the Attribute Editor, open the **Shadows** section and click on **Use Depth Map Shadows**.

  Leave the default settings. They will be explained in more depth in a later lesson.

2 Set Render Globals

The Render Globals are a set of global attributes that you can set to define how your scene will render. To set up the quality of the test rendering, you need to set the Render Globals.

- In the Render View panel, click with your right mouse button and choose **Options → Render Globals...**
- Open up the **Anti-aliasing Quality** section.
- Set the presets to **Intermediate Quality**.

  Anti-aliasing is a visual smoothing of edge lines in the final rendered image. Because bitmaps are made up of square pixels, a diagonal line would appear jagged unless it was somehow anti-aliased.

- Close the Render Globals window

3 Display resolution gate

Your current view panel may not be displaying the actual proportions that will be rendered. You can display the camera's resolution gate to see how the scene will actually render.

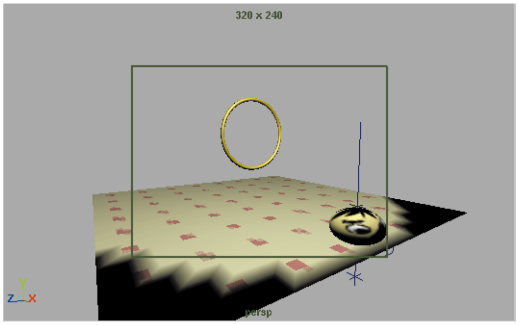
- Make the Perspective view the active panel.
- Use the hotbox to select **View → Camera Settings → Resolution Gate**.

  The view is adjusted to show a bounding box that defines how the default render resolution of 320 x 240 pixels relates to the current view.

- Dolly into the view so that it is well composed within the resolution gate. Try to set up a view where the ball starts in the back of the scene then animates to the front.

Keep in mind that only objects within the box will be rendered.

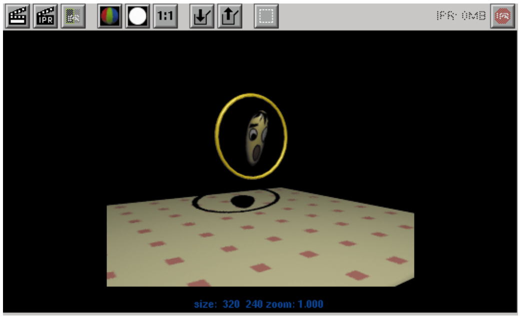
- Playback the animation to confirm that the scene works for the whole sequence. If not, then edit the view until you are happy with the camera setup.



New view

4 Test render the scene

- Select **Options → Time Slider**.
- Go to frame 30.
- In the Render View panel, click with the right mouse button and choose **Render → Render → persp**.



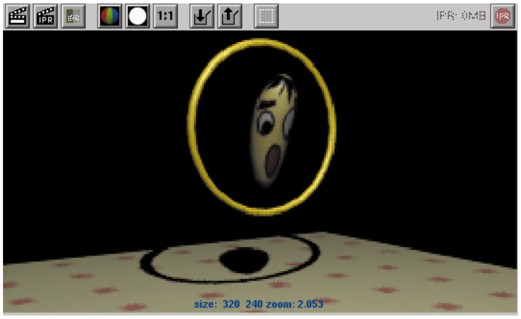
Rendered scene

5 Zoom into the rendering

You can view the rendering using the **Alt** key and the zoom and track hotkeys.

- Use the **Alt** key and the left and middle mouse buttons to zoom in to the view.

  Now you can evaluate in more detail how your rendering looks at the pixel level.



Close-up of rendering

- In the Render View panel, click with your right mouse button and choose **View → Real Size.**

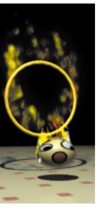
Rendering animations

Once you are happy with your test rendering, it is time to render an animation. This will be accomplished using Maya's *batch renderer*. In preparation, you will add motion blur to your scene to simulate the blur generated in live action film and video work.

1 Set the Image output

To render an animation, you must set up the scene's file extensions to indicate a rendered sequence. You must also set up the Start and End Frames.

- From the Hotbox, select **Window → Render Globals...**



- From the **Image File Output** section, set the following:

  File Name Prefix to **bounce**;

  This sets the name of the animated sequence.

  Frame/Animation Ext to:

  name.#.ext (for NT)

  name.ext.# (for IRIX);

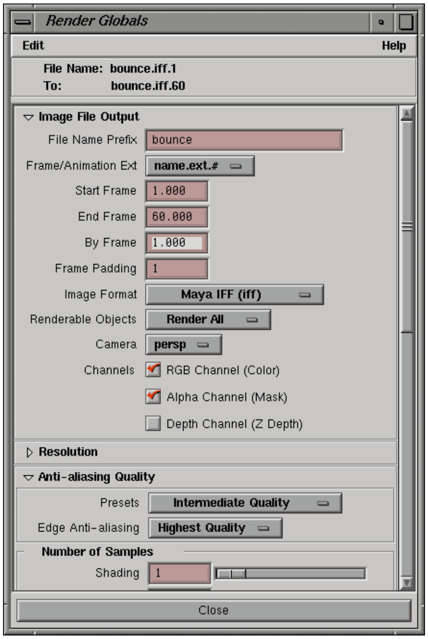
  This sets Maya up to render a numbered sequence of images.

  Start Frame to **1**;

  End Frame to **60**;

  By Frame to **1**.

  This tells Maya to render every frame from 1 to 60.



Render Globals

Edit Help

File Name: bounce.iff.1
To: bounce.iff.60

Image File Output

File Name Prefix: bounce
Frame/Animation Ext: name.ext.#
Start Frame: 1.000
End Frame: 60.000
By Frame: 1.000
Frame Padding: 1
Image Format: Maya IFF (iff)
Renderable Objects: Render All
Camera: persp
Channels: ☑ RGB Channel (Color)
☑ Alpha Channel (Mask)
□ Depth Channel (Z Depth)

Resolution

Anti-aliasing Quality

Presets: Intermediate Quality
Edge Anti-aliasing: Highest Quality

Number of Samples

Shading: 1

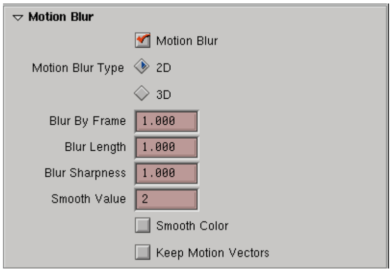
Close

Render Globals

2 Turn on motion blur

- In the Render Globals **Motion Blur** section, click on the **Motion Blur** button to turn it on.
- Set the Motion Blur Type to be 2D.

  This type of motion blur renders the fastest.



Motion Blur

☑ Motion Blur
Motion Blur Type: ◆ 2D
◇ 3D
Blur By Frame: 1.000
Blur Length: 1.000
Blur Sharpness: 1.000
Smooth Value: 2
□ Smooth Color
□ Keep Motion Vectors

Motion blur check box

3 Batch render the scene

- Press **F5** to change to the **Render** menu set.
- Use the hotbox to select **Render → (Save) Batch Render...**
- Enter the name *bounce_anim* next to the path.

  You will notice that this file is being saved into the *render* directory instead of the *scenes* directory.
- Click on the **Save/Render** button.

4 Watch the render progress

The sequence will be rendered as a series of frames.

- Use the hotbox to select **Window → General Editors → Script Editor.**

In this window, you can watch a series of status entries for your animation.

5 View the resulting animation

After the rendering is complete, you can preview the results using the *fcheck* utility.

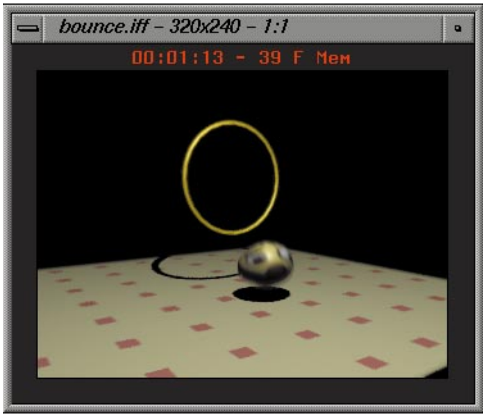
On NT

- Open the fcheck utility by clicking on its icon.
- Select **File → Open Animation**.
- Navigate to the `projectOne\images` folder.
- Select one of the frames of your animation then click **Open**.

On IRIX

- In a shell window, set your current directory to the `maya/projects/learningMaya/projectOne/images` directory.
- Type the following:

```
fcheck bounce.iff.
```



Animation previewed with fcheck utility

In both cases, the animation will load one frame at a time then playback more quickly once in memory.

Tip: To learn more about the capabilities of fcheck for previewing your animations, enter `fcheck -h` in a shell window.

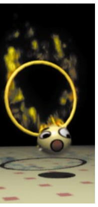
6 Save your work

- From the **File** menu, select **Save Scene As...**
- Enter the name *bounce_03* next to the file's path.
- Press the **Save** button or press the **Enter** key.

Conclusion

You have now touched some of the basic concepts in texturing and rendering a scene. Maya's shading groups offer a lot of depth for creating the look of your scenes. In later lessons, you will learn how to make use of some of this power.

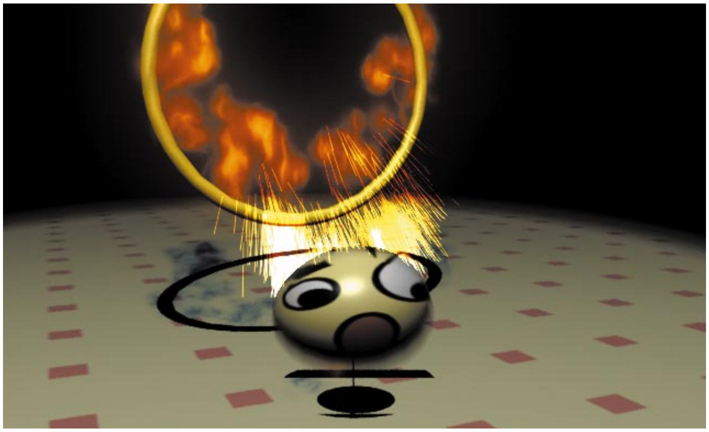
The resulting animation shows a ball with a texture map of a face. In the next lesson, you will use particles to create a ring of fire that helps explain the concerned look on the ball's face.



4 Particles

Particles are small object types that can be animated using dynamic forces in place of traditional keyframes. These effects are in essence *simulations* of physical effects such as water, smoke and fire.

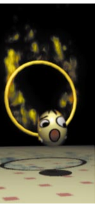
To add particle effects to the scene, you are going to add flames to the ring. These flames will be created using Maya's default particle fire effect. You will then create sparks that will be animated to appear when the ball touches the fire.



The ring of fire

In this lesson you will learn the following:

- How to add a fire effect to a NURBS torus
- How to collide the ball and the flames
- How to set up a particle event
- How to define a particle attribute using a ramp
- How to define a particle attribute using an expression
- How to software render a particle animation
- How to hardware render a particle animation



Project set-up

If you are continuing from the last lesson, then you can begin working right away. If not, then open the Maya file you saved in that lesson. You will continue to use the menu-less user interface to explore the use of hotkeys and the hotbox.

1 Display the Time slider

To work with particles, you will need the Time slider.

- From the hotbox, select **Options** → **Time Slider**.

2 Set up your Perspective panel

To simplify the workspace, you will focus on a single Perspective view. You will also turn off the resolution gate to let you focus on the particles.

- Use the spacebar to make the Perspective view panel full screen.
- Select **View** → **Camera Settings** → **No Gate** to turn off the resolution gate.

3 Change menu sets

- Press **h** and with the left mouse button, choose **Dynamics** from the marking menu.

Note: You could have also used the **F4** hotkey to change menu sets.

Ring of Fire

Using one of Maya's preset particle effects, you will add fire to the ring. This preset creates everything needed to make the particles act and look like fire.

1 Adding the Fire Effect

- Select the *ring* torus.
- Select **Effects** → **Create Fire**.

2 Playback the simulation

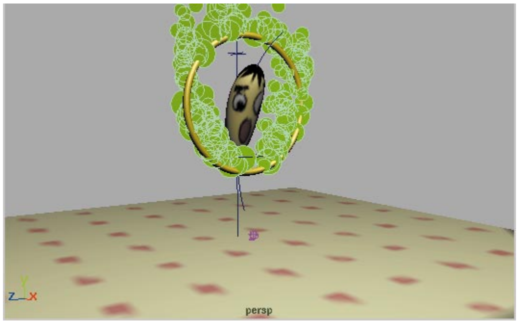
- Click the **Animation Preferences** button found at the right side of the Time slider.

  If this icon is not visible then select **Options** → **Range Slider** to show the icon.
- In the **Playback** section, set the following:

  Playback Speed to **Free.**

  When working with particles it is *very important* that the playback speed is set to free. Otherwise your simulations may act unpredictably.
- Click the **Save** button.
- Go back to frame 1 and playback the simulation.

  The particles are generated from the ring. The simulation plays back at the same time the ball is animating. In Maya, you can combine animated objects with your particle simulations.



Default fire particles

The fire effect is the result of a particle object that is controlled by several dynamic fields such as gravity and turbulence. The fire preset added these elements to your scene and lets you easily control them.

Note: Particle simulations should always start at frame 1 for you to see accurate results.

3 Editing the fire attributes

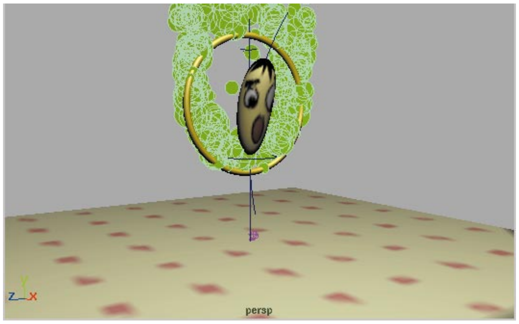
To control the various parts of the fire effect, you can simply edit attributes that are designed specifically for the fire effect.

- Playback the simulation to a point where particles are visible then stop.
- Select the fire particles.
- In the Attribute editor, make sure that the *particleShape* tab is pressed.
- Rename the particles *flame*.
- Scroll down to the bottom and open the **Extra Attributes** section.
- Set the following attributes:

  Fire Density to **15**;

  Fire Lifespan to **1.5**.
- Go back to frame 1 and playback the simulation.

  Now there are more fire particles and they last longer.



Updated fire particles

4 Setting the Initial state

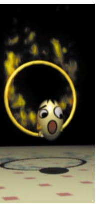
One problem you may notice with the simulation is that there are no particles when the animation starts. Since the ring of fire needs to be lit at all times, you must set the particle's initial state.

- Playback the scene until around frame 30 then stop playback.
- Select the particles.
- From the **Solvers** menu, select **Initial State → Set for Selected**.
- Go back to frame 1 and playback the simulation.

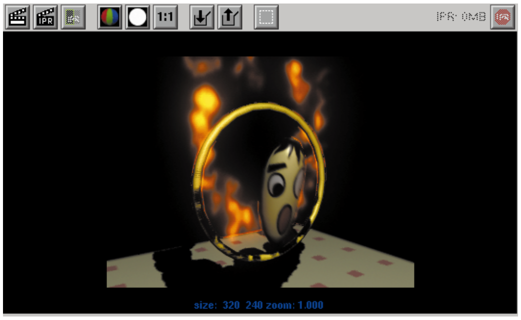
  By setting the initial state for the particles, you can see that at frame 1, the particles are already created.

5 Test render the particles

- Playback the scene until around frame 35 then stop playback.
- Press **F5** to go to the **Rendering** menu set.
- From the **Render** menu, select **Render into New Window...**



The scene is now rendered with the fire particles included. Some particles can be rendered using the software renderer which allows them to be automatically integrated into the scene.



Software rendering

Note: The flame's shadows don't look very convincing because the depth map shadows don't work well with the fire's volumetric shader. Also, there is a strange halo around the fire that is caused by the 2D motion blur. You will fix these problems later in the lesson.

Sparks

As an added effect, you will set up more particles that will represent emitted sparks when the ball touches the flames. You will set up this effect to occur at exactly the point where the touches hits the flames.

1 Set up particle collisions

To make the ball and the particles collide, you must define them as colliding objects.

- **Select** the *flame* particles.
- Press the **Shift** key then **select** the ball.

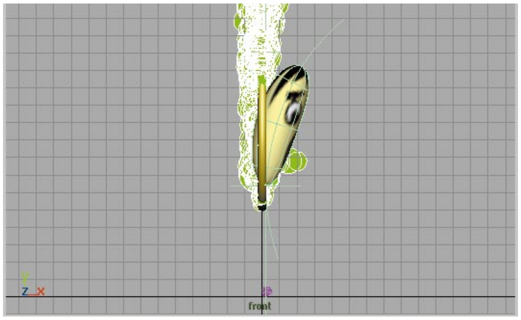
Note: It is very important that the ball is selected last.

- Press **F4** to go back to the **Dynamics** menu set then from the **Particles** menu, select **Make Collide**.

2 Preview the results

- Select **Panels** → **Orthographic** → **front**.
- Playback the simulation.

  When the ball collides with the flames they are pushed a little forward.



Minimal collisions

Note: If you are not getting a strong collision, you may want to adjust the ball's Y translate animation curve in the graph editor so that it jumps just over the bottom of the ring.

3 Adding friction

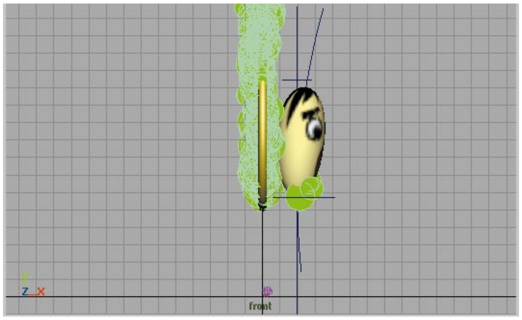
- **Select** the fire particles on their own.

- In the Attribute editor, click on the *geoConnector* tab.
- Set the following attributes:

  Resilience to **0.3**;

  Friction to **0.9**.
- Playback the simulation.

  Now the flames react a little more when they collide with the ball. The lower resilience allows them to react more and the friction is allowing the ball to drag the flames forward.



Stronger collisions

4 Create a Particle Event

To generate the sparks, you will use the collision between the ball and the flames to emit a new particle object.

- **Select** the fire particles.
- From the **Particles** menu, select **Particle Collision Events...**
- In the Particle Collision Events window, go to the Event Type section and set the following.

  Type to **Emit**;

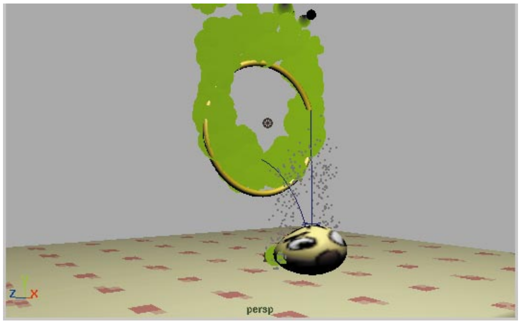
  Random # Particles to **On;**

  Num particles to **6**;

  Spread to **0.2**;

  Inherit Velocity to **0.8**.
- Click **Create Event**.
- Playback the simulation.

  Several small particles are emitted after the ball touches the flames. These particles float forward based on the momentum they received from the collision. Now you will adjust how they react and how they look.



Particle collision event

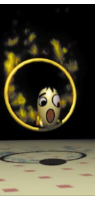
5 Add Gravity to the particles

- Select the new particles.
- From the **Fields** menu, select **Create Gravity**.
- In the Attribute editor, set the **Magnitude** to **25**.

  This will make them drop more quickly.

- Playback the simulation.

  Now the particles drop to the ground after the collision. The gravity field is pulling them down.



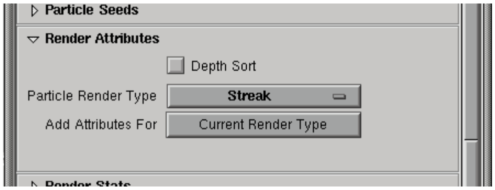
Designing the look of the sparks

To create particles that look like sparks, you need to adjust various particle attributes. In this case, you will create particles that will be streaks that die fairly quickly after being emitted. Their color will start out white, and then turn to a yellow-orange.

1 Change Render Type to streak

Particles can have their render type set from a list of possible looks. You can switch between the different types until you get one that suits your needs.

- Select the new particles.
- Rename them *sparks*.
- In the Attribute Editor, go to the **Render Attributes** section and set **Render Type** to **Streak**.



▷ Particle Seeds

▽ Render Attributes

□ Depth Sort

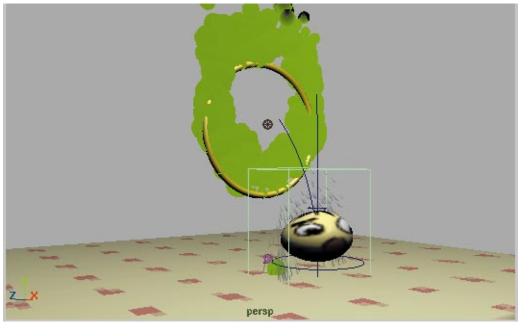
Particle Render Type: Streak

Add Attributes For: Current Render Type

▷ Render Stats

Render Attributes

This render type is designed to work with hardware rendering. This means that later, you will have to composite the final hardware rendered particles with software rendered scenes.



Streak particles

2 Add and edit render attributes

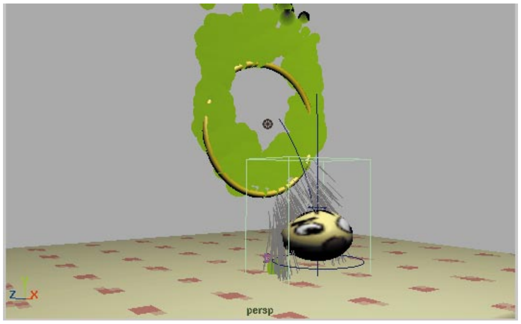
- Click on the **Current Render Type** button.
- Set the **Render Attributes** as follows:

  Line Width to **1**;

  Tail Fade to **0.67**;

  Tail Size to **3.0**.

  This gives the sparks a much stronger presence. The higher tail fade value means that it doesn't taper off too strongly and the tail size value lengthens the sparks.



Streak particles

3 Add color per particle

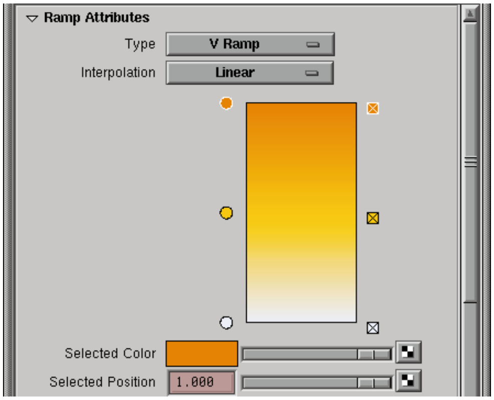
The particle node has the ability to have new attributes added to it as they are needed. This lets you add complexity to a particle node when you need to.

You can use this technique to add color to the particles that will affect the particles individually instead of as a group.

- In the **Add Dynamic Attributes** section, click on the **Color** button.
- From the Particle color window, select **Add Per Particle Attribute** then click the **Add Attribute** button.

  This adds a *rgbPP* line to the **Per Particle Attributes** section.
- Click on the *rgbPP* field with your right mouse button and select **Create Ramp**.
- Click again on the *rgbPP* field with your right mouse button and select **<-arrayMapper.outColorPP → Edit Ramp**.

  In the Ramp window, you will find three markers each with a square and a circular icon.
- Click on the circle icon at the bottom of the ramp then click on the color swatch next to **Selected Color**.
- Change the color to white.
- Complete the same steps to change the middle marker to a yellow and the top marker to a yellow-orange.



Ramp Attributes
Type: V Ramp
Interpolation: Linear
Selected Color
Selected Position: 1.000

Particle color ramp

- Playback the simulation.

  Now the particles start out white then become yellow and orange over time.

4 Particle Lifespan

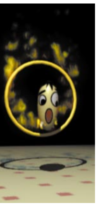
The Lifespan attribute lets you determine how long the particle will remain in the scene before it disappears or dies. In this case you will add the lifespan on a per particle basis then use an expression to add randomness to the particles.

- With *sparks* selected, go to the **Add Dynamic Attributes** section in the Attribute editor.
- Click the **Lifespan** button.

  A window pops up offering you the chance to add this attribute on a per object basis or a per particle basis.
- Select **Add Per Particle Attribute**.
- Click the **Add Attribute** button.

5 Add a Creation Expression

This adds a LifespanPP attribute to the particle node's Per Particle section. By



selecting Add Per Particle Attribute, you can set a different value for each of the particles. For lifespan you can use a simple mathematical expression to add randomness to the simulation.

- Click on the *lifeSpanPP* field with your right mouse button and select **Creation Expression**.

  This opens the Expression editor.

- In the Expression field enter the following;

  ```
  lifespanPP = rand (0.05, 0.55)
  ```

  This expression sets a random number between 0.05 and 0.55 for the lifespan of each particle.

- Click the Create button,
- Playback the simulation.

  Now the particles last for somewhere between 0.05 and 0.55 seconds. This gives the sparks a more random look.

6 Save your work

- From the **File** menu, select **Save Scene As...**
- Enter the name *my_ball4* next to the file's path.
- Press the **Save** button or press the **Enter** key.

Rendering Particles

It was discussed earlier that the sparks used a particle type that could only be rendered using hardware rendering while the fire used software rendering. The question, therefore, is how do you bring hardware rendered particles together with software rendered scene?

The answer is to render them separately, and then bring them together using a compositing package such as *Maya Composer* (IRIX) or *Maya Fusion* (NT).

To composite the particles with the ball, you will need to render the top layer – in this case the sparks – with a matte, or *mask*.

The mask is a grayscale channel that defines what areas are to be masked out when layered on top of the background. In this scene, the background is the bouncing ball, the ring of fire, and the floor.

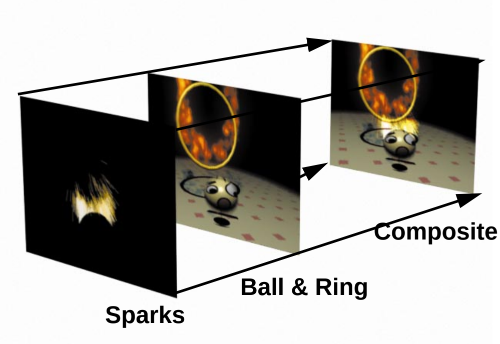


Diagram of compositing layers

Software rendering

The flames created using the fire effect can be rendered using software rendering. This means creating another batch rendering of your scene. This will represent the first render pass that can then be later composited together with the sparks.

1 Change your motion blur type

Since the 2D motion blur used in the last lesson doesn't render well with the flame particles, you will switch to the 3D motion blur type.

- Select **Window** → **Render Globals**.

- Open the **Motion Blur** section and change the **Motion Blur Type** to **3D**.

  This type of motion blur renders more slowly but is more accurate and works better with software rendered particles.

2 Fix the flame shadows

Earlier it was noted that the shadows generated from the flames didn't look correct. The depth map shadows can not recognize the subtleties of the volumetric shader used for the flames and raytrace shadows are needed.

- **Select** the spotlight that is illuminating the scene.
- In the Attribute editor, open the **Shadows** section, scroll down to **Raytrace Shadow Attributes** and set **Use Ray Trace Shadows** to **On**.

  This sets up the light, but to use raytraced shadows, you will need to turn on raytracing itself.
- Open the Render Globals.
- Open the **Raytracing Quality** section and turn **Raytracing** to **On**.

Note: Maya uses a selective raytracer and only objects that require reflections, refractions or Raytrace shadows will use this technique.

3 Limiting the Reflections

When Raytracing is turned on, any shader that has a reflectivity value will render with reflections. It is therefore a good idea to set Reflectivity to 0 for some objects so that the rendering be as fast as possible.

- Open up a Hypershade panel.
- **Select** the ball, the floor and the ring in the perspective view.
- In the Hypershade's **Graph** menu, select **Graph Materials on Selected Objects**.
- Select the *ballM* material node.
- In the Attribute editor, go to the Specular shading section and set **Reflectivity** to **0**.

  Now the ball won't reflect anything.
- Set **Reflectivity** to **0** for the *ring*.
- Set **Reflectivity** to **0.33** for the *floor* so that you do see some reflectivity in the scene.

4 Re-size of the floor surface

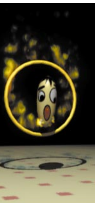
To make the floor larger so that it extends beyond the spotlight's influence, you will scale it out then update its texture placement to match the new size.

- Select the *floor* surface.
- In the Attribute editor, set the following:

  Scale X to **4**;

  Scale Y to **4**;

  Scale Z to **4**.
- In the Hypershade, select the *floorM* shader.
- Click on the **Show upstream connections** button.
- Click on the *place2Dtexture* node.



- In the Attribute editor, set the following:

  Repeat U to **32**;

  Repeat V to **32**.

5 Batch render the scene

- Press **F5** to change to the **Render** menu set.
- Use the hotbox to select **Render** → **Batch Render...**
- Enter then name *fire_bounce* next to the path.
- Click on the **Batch Render** button.

  This will create a render pass that includes the geometry and the fire. Now you will render the sparks using hardware rendering.

Hardware rendering

You have been using hardware rendering in the Perspective view panel to help preview the scene. You can also use hardware rendering to render the spark particles so that they match the rendered scene.

1 Set the opacity of the fire

Since you only want the sparks to appear in the hardware rendering, you will need to hide the flames by setting an opacity value to 0.

- Select the *flame* particles.
- In the Attribute editor, scroll down to the **Add Dynamic Attributes** section and click on **Opacity**.
- Choose **Add Per Object Attribute** then click on the **Add Attribute** button.
- This adds an opacity attribute to the **Render Attributes** section.
- Set **Opacity** to **0**.

  Now the particles will not be visible in the hardware rendering. You can set opacity to 0 when you want to hardware render the scene and to 1 when you want to software render the scene.

2 Set the hardware render attributes

- Select **Window** → **Rendering Editors** → **Hardware Render Buffer**...
- In the Hardware Render Buffer window, select **Render** → **Attributes**...
- In the Attribute editor, set the following attributes:

  Filename to *sparks*;

  Extension to **name.1.ext** (NT)

  or to **name.ext.1** (IRIX)

  This should match the extension setting you chose for your software rendering.

  Start Frame to **1**;

  End Frame to **60**;

  Resolution to **320x240**;

  This sets the length and size of the animation.

  Alpha Source to **Luminance**;

  Geometry Mask to **On**.

  This will use the ring, floor and the ball as mask objects and the particle luminance to define the alpha. An alpha channel, otherwise known as a matte channel, is important for layering images in a compositing package later.

- Open the **Multi-Pass Render options** section and turn

  Multi-Pass rendering to **On**;

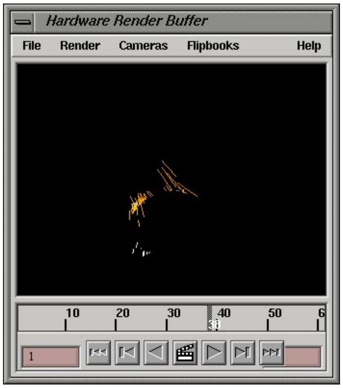
  Anti-Alias Polygons to **On**.

  This will soften the look of the rendering.

Note: In some ways, these attributes are similar to the Render Globals, except that these settings only affect hardware rendering.

3 Test a frame

- Playback the simulation until you hit a frame where some of the sparks appear.
- Click on the **Test** button in the middle of the Render Buffer's time controls.



Hardware render buffer

Tip: The resolution gate must be off when performing a hardware render if it is to be composited with the batch render.

4 Render a sequence

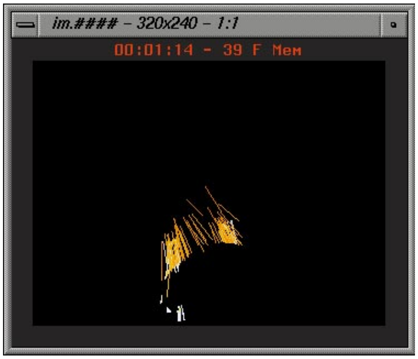
You can now render a whole animation using this window. Compared to software rendering, this window lets you use the speed of hardware rendering to generate animations quickly.

- Select **Render** → **Render Sequence**.

5 Preview the resulting flipbook

Once the sequence is complete, you can preview the results.

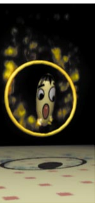
- Select **Flipbooks** → **sparks.1-60**

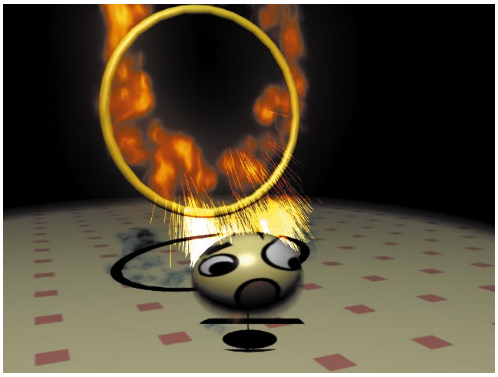


Animation preview

6 Composite rendered animations

You now have an animation of a bouncing ball and a sequence of particles embedded with an alpha channel. You can now use your compositing software to layer all the elements together.





Final composite

There are several advantages to compositing your layers instead of rendering them all into one scene:

- by separating background and foreground elements and rendering them individually, rendering times can be greatly reduced;
- by rendering different elements on different layers, it is easier to later make revisions to one layer without having to re-render the whole scene;
- when working with particles, interesting effects can also be achieved by compositing hardware and software rendered particles; and
- by using different layers, you can use your compositing software to adjust the color for a particular layer without affecting other layers.

Conclusion

Congratulations, you now have a complete scene that includes objects, shaders and visual effects. You have also begun to develop skills that you will use throughout your use of Maya.

The next chapter is a more in-depth look into some of the user interface elements that you have been using in these lessons. Once you have read this chapter, you will be able to make your own decisions on how you want to configure the UI for your needs.

In the projects that follow, the instructions will not specify whether or not you should use the hotbox or menus to complete an action. The choice will be left up to you.

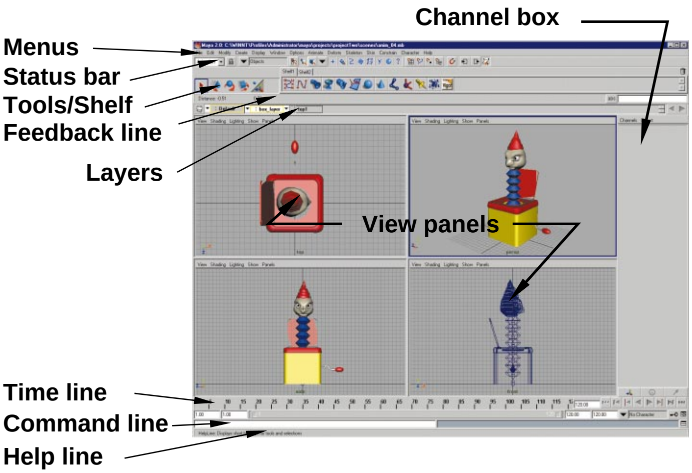
5 Working with Maya

If you have just completed the first four tutorials, then you have worked with Maya from modeling and animating to rendering and particles. Now is a good time to review some of the user interface concepts that you have worked with to give you a more complete overview of how Maya works.

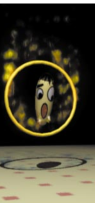
It is recommended that you work through this chapter before proceeding with Learning Maya. This chapter explores the basic user interface actions that you will use in your day-to-day work.

The Workspace

You have learned how to build and animate scenes using different view panels and user interface tools. The panels offer different points of view for evaluating your work – such as perspective views, orthographic views, graphs and Outliners – while the tools offer you different methods for interacting with the objects in your scene. Shown below is the workspace and its key elements:

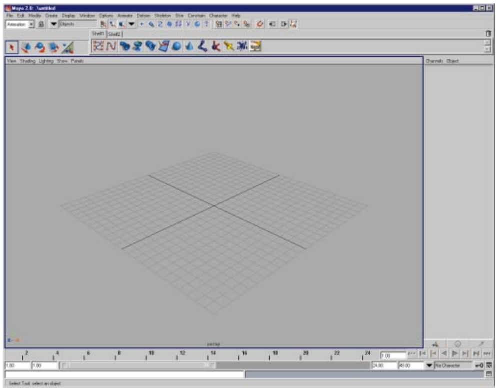


The Maya workspace



Layouts

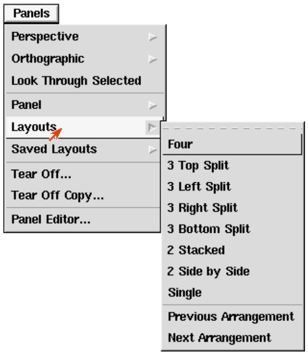
When Maya is first launched, you are presented with a single Perspective view panel. As you work, you may want to change to other view layouts.



The default layout

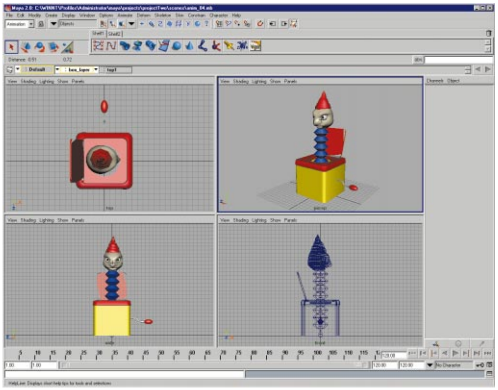
To change your view layouts:

- Go to the view panel's **Panels** menu and select a new layout option from the **Layouts** pop-up.



The Layouts pop-up menu

You can set up various types of layouts ranging from two to four panels.



A four view layout

Tip: If you are looking at several view panels at the same time and you want to focus on one of them, tap the space bar and the view will become full screen. Tap the spacebar again and the panels will return to the previous layout.

View panels

As you begin to build and animate objects, you will want to view the results from various points of view. In Maya, you can place either perspective or orthographic views in each panel.

To change the content of a view panel:

- Go to the view panel's **Panels** menu and select a view type from either the **Perspective** or **Orthographic** pop-ups.

View tools

When you are working with perspective and orthographic views, you can change your view point by using hotkey view tools.

To tumble in a Perspective view:

- Press the **Alt** key and click-drag with the left mouse button.

Tip: The ability to tumble an orthographic view is locked by default. To unlock this feature, you need to select the desired orthographic view and open the Attribute Editor.

To track in any view panel:

- Press the **Alt** key and click-drag with the middle mouse button.

To dolly in any view panel:

- Press the **Alt** key and click-drag with both the left and middle mouse buttons.

These view tools allow you to quickly work in 3D space using a simple hotkey.

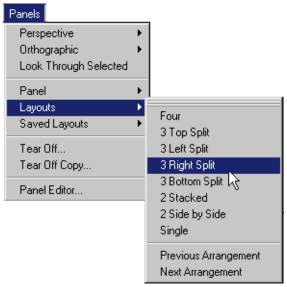
Tip: You can also track and dolly in other view panels such as the Hypergraph, the Graph Editor, Visor, Hypershade, and even the Render View window. The same view tools work for most panel types.

Other panel types

You can also change the content of the view panel to display other types of information such as the Multilister or the Graph Editor.

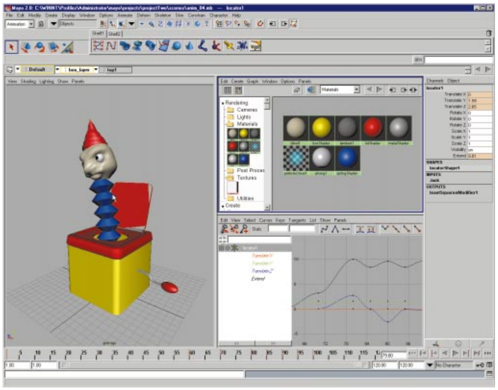
To change the content of a view panel:

- Go to the view panel's **Panels** menu and select a panel type from the **Panel** pop-up.



The Panel pop-up menu

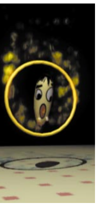
In the workspace below, you can see a Hypershade panel for helping you organize your shading groups and a Graph Editor for working with animation curves.



The workspace with various panel types

Saved layouts

As you become more familiar with Maya, you may want to set up an arrangement of panels to suit a particular workflow. For example, you may want a Dope sheet, a Perspective view, a Top view and a Hypergraph view all set up in a particular manner.



To add a new layout of your own:

- Go to the view panel's **Panels** menu and select **Saved Layouts** → **Edit Layouts...**

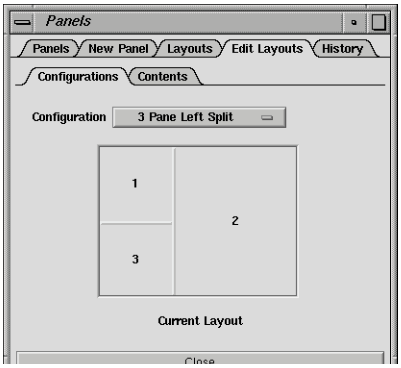
  In the Edit window, you can add a new saved layout and edit the various aspects of the layout.

To add a new layout to the list:

- Click on **New Layout**.
- Select and edit the layouts name.
- Press the **Enter** key.

To edit the configuration of a saved layout:

- Press the **Edit Layouts** tab.
- Choose a configuration, then click-drag on the center bars to edit the layout.



Configuration 3 Pane Left Split

Layout Editor

- Press the **Contents** tab.
- Choose a panel type for each of the panels set up in the configuration section.

Display options

Using the Shading menu on each view panel, you can choose which kind of display you want for your geometry.

To change your panel display:

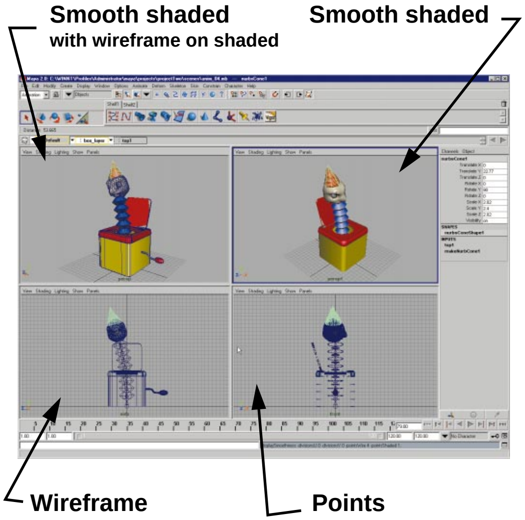
- Go to the panel's **Shading** menu and select one of the options.

Or

- Make the panel you wish to change the active panel.
- Use one of the following hotkeys to switch display types:

  4 for wireframe;

  5 for smooth shaded.



Various display styles

Texturing and lighting

Another important option found on this menu is hardware texturing. This option allows you to visualize textures interactively in the view panels – but you require texture RAM on your computer to use this feature.

To use hardware texturing:

- Build a shader that uses textures.
- Go to the panel’s **Shading** menu and select **Hardware Texturing**.

Or

- Press the following hotkey:

  6 for hardware texturing.

To display different textures:

It is possible to display texture different texture maps on your surface during hardware texturing. For instance, you could display the color map or the bump map.

- Select the material that is assigned to your objects.
- In the Attribute editor, open the **Hardware Texturing** section and set the **Textured channel** to the desired channel.
- You can also set the **Texture quality** for each material node.

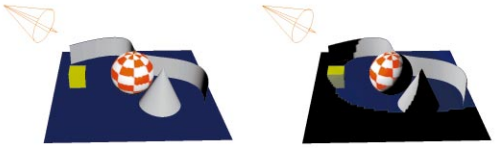
To add hardware lighting to your scene:

- Add a light into your scene.
- Go to the panel’s **Lighting** menu and select one of the options.

Or

- Press the following hotkey:

  7 for realistic lighting.



Hardware lighting and texturing

Display smoothness

By default, NURBS surfaces are displayed using a rough smoothness setting in order to enhance playback and interactivity of scenes.

To increase objects smoothness:

- Go to the **Display** menu and under **NURBS Smoothness** choose one of the options.

Or

- Use one of the following hotkeys to switch display types:

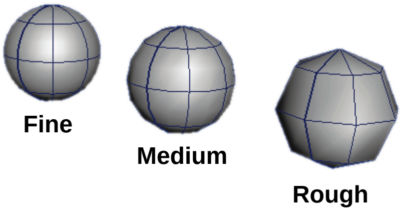
  1 for rough;

  2 for medium;

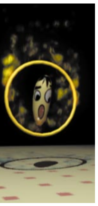
  3 for fine.

Or

- Press the **d** key and choose an option from the marking menu.



Surface smoothness

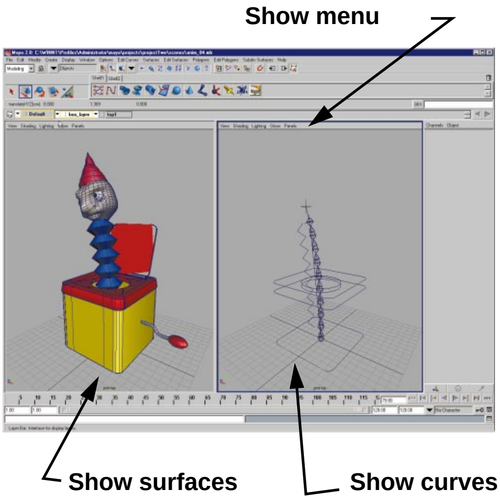


Tip: If you want to set a particular smoothness when you first place a piece of geometry into the scene, go to the **Options** menu and select **General Preferences**. In the **Display** section, you have various options for setting defaults.

Show menu

The Show menu is an important tool found on each view panel's menu. This menu lets you restrict what each panel can show on a panel-by-panel basis.

Restricting what each panel shows, lets you display curves in one window and surfaces in another to help edit construction history. Or you can hide curves when playing back a motion path animation while editing the same curve in another panel.



The Show menu

UI preferences

The Maya workspace is made up of various user interface elements which assist you in your day-to-day work. By default, this interface puts all of these on the screen for easy access.

To reduce the user interface to only view panels and menus:

- Go to the **Options** menu and select **Show Only Viewing Panes**.

With less user interface clutter, you can rely more on hotkeys and other user interface methods for accessing tools while conserving screen real-estate.

To go back to a full user interface:

- Go to the **Options** menu and select **Show All Panes**.

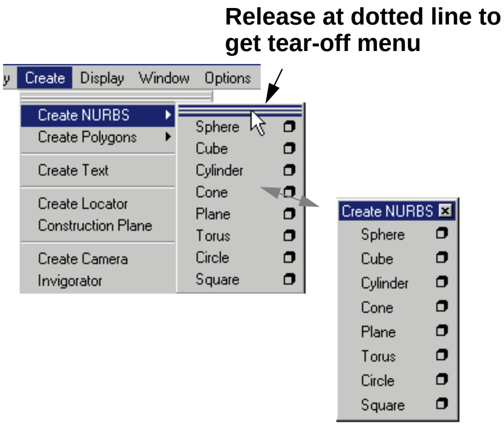
Menus

Most of the tools and actions you will use in Maya are found in the main menus. The first seven menus are always visible, while the next few menus change depending on which UI mode you are in.

Menus and menu pop-ups that display a dotted line at the top can be 'torn off' for easier access.

To tear off a menu:

- Pull down on the menu then release as your mouse hits the top of the menu.



A tear-off menu

Later in this chapter, you will look at the difference between the two main types of menu items – *tools* and *actions*.

Menu sets

There are four menu sets in Maya: *Animation*, *Modeling*, *Dynamics* and *Rendering*. These allow you to focus on tools appropriate to a particular workflow.

To choose a menu set:

- Select the menu set from the pop-up menu found at the left of the Status Line bar.

To choose a menu set using hotkeys:

- Press the **h** key and choose the desired UI mode from the radial marking menu.

To choose a menu set using function keys:

- Press **F2** for Animation;
- Press **F3** for Modeling;
- Press **F4** for Dynamics;
- Press **F5** for Rendering.

The shelf

Another way of accessing tools and actions is using the shelf. You can move items from the menu to the shelf to begin combining tools into groups based on your personal workflow needs.

To add a menu item to a shelf:

- Press **Ctrl**+**Alt**+**Shift** then select the menu item. It will appear on the active shelf.

To edit the shelf contents and tabs:

- Go to the **Options** menu and select **Customize UI → Shelves...**

Hotbox

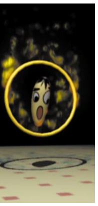
As you have learned, pressing the spacebar quickly pops a pane between full screen and its regular size. If you press a little longer, you gain access to the hotbox.

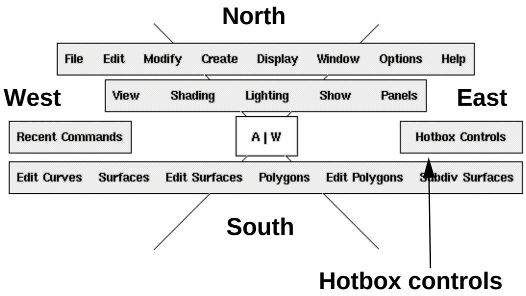
The hotbox is a user interface tool that gives you access to as much or as little of the Maya UI as you want. It appears where your cursor is located and offers the fastest access to tools and actions.

To access the hotbox:

- Press and hold the spacebar.

PROJECT ONE

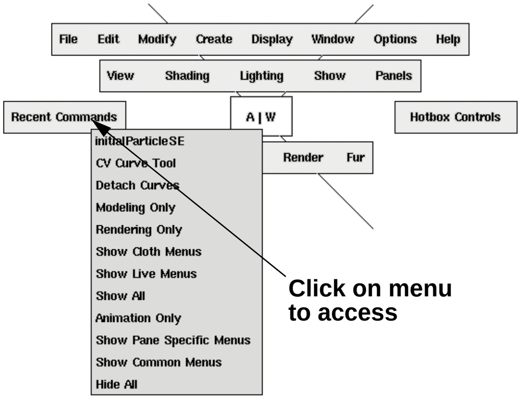



North
File Edit Modify Create Display Window Options Help
West
View Shading Lighting Show Panels
East
Recent Commands
A | W
Hotbox Controls
Edit Curves Surfaces Edit Surfaces Polygons Edit Polygons Subdiv Surfaces
South
Hotbox controls

The hotbox with four quadrants marked

The hotbox offers a fully customizable interface element that provides you with access to all of the main menus as well as your own set of marking menus.

You can use the **Hotbox Controls** to display or show as many or as few menus as you need, and you can configure up to 15 different marking menus for fast interaction.



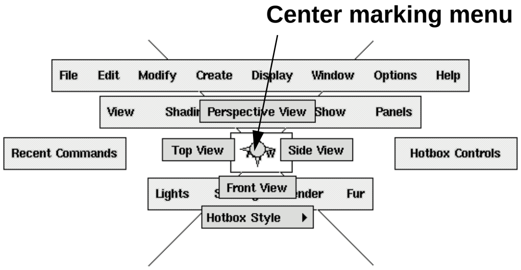
Accessing the recent commands menu

Hotbox marking menus

You can access marking menus in five areas of the hotbox. Since each of these areas can have a marking menu for each mouse button, it is possible to have 15 menus in total. You can edit the content of the marking menus by going to the **Options** menu and selecting **Customize UI → Marking menus**.

To access the center marking menu:

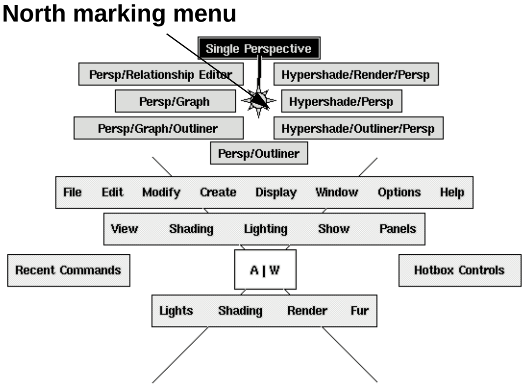
- Press the spacebar.
- Click and drag in the center area to access the desired menu.



The center marking menu

To access the edge marking menus:

- Press the spacebar.
- Click and drag in the top quadrant to access the desired menu.

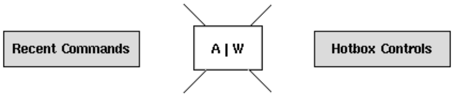

North marking menu
Single Perspective
Persp/Relationship Editor
Hypershade/Render/Persp
Persp/Graph
Hypershade/Persp
Persp/Graph/Outliner
Hypershade/Outliner/Persp
Persp/Outliner
File Edit Modify Create Display Window Options Help
View Shading Lighting Show Panels
Recent Commands
A | W
Hotbox Controls
Lights Shading Render Fur

A quadrant-based marking menu

Customizing the hotbox

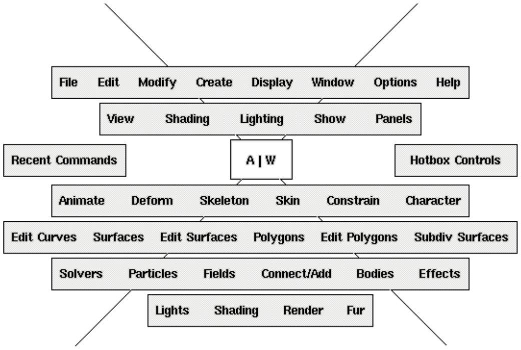
You can customize the hotbox to make it as simple or as complex as you need. You can choose which menus are available and which are not.

If you want, you can reduce the hotbox to its essentials and focus on its marking menu capabilities.



A reduced hotbox layout

Alternatively, you could hide the other UI elements, such as panel menus, and use the hotbox for access to everything. You get to choose which method works best for you.


File Edit Modify Create Display Window Options Help
View Shading Lighting Show Panels
Recent Commands
A | W
Hotbox Controls
Animate Deform Skeleton Skin Constrain Character
Edit Curves Surfaces Edit Surfaces Polygons Edit Polygons Subdiv Surfaces
Solvers Particles Fields Connect/Add Bodies Effects
Lights Shading Render Fur

A complete hotbox layout

To customize the hotbox:

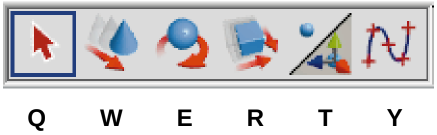
- Use the Hotbox controls.

Or

- Use the center marking menu.
- Choose an option from the Hotbox Styles menu.

Tool manipulators

At the top left of the workspace you have access to important tools. These include the **Select**, **Move**, **Rotate**, **Scale** and **Show manipulator** tools. Each of these is laid out to correspond to a related hotkey that can be easily remembered using the QWERTY keys on your keyboard.



The QWERTY tool layout

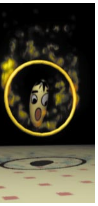
These tools will be used for your most common tool-based actions—like selecting and transforming.

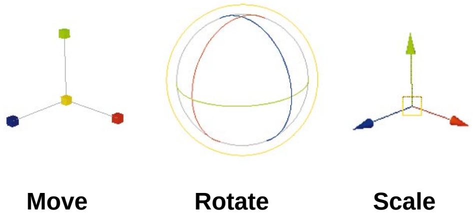
Note: The Y key drives the last spot on the QWERTY palette which is for the last tool used. The advantages of this will be discussed later in the *Tools and Actions* chapter of this document.

Transform manipulators

One of the most basic node types in Maya is the *transform node*. This node contains attributes focused on the position, orientation and scale of an object. To help you interactively manipulate these nodes, there are three transform manipulators which make it easy to constrain along the main axes.

Each of the manipulators uses a color to indicate their axes. RGB is used to correspond to XYZ. Therefore, red is for X, green for Y and blue for Z. Selected handles are displayed in yellow.





Transform manipulators

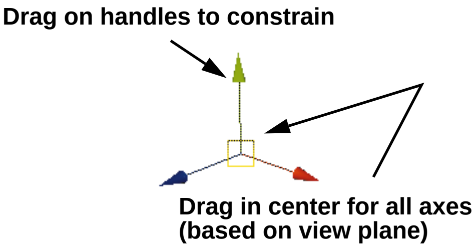
To explore some of the options available with manipulators, you will use the transform manipulator.

To use a transform manipulator in view plane:

- Click-drag on the center of the manipulator.

To constrain a manipulator along one axis:

- Click-drag on one of the manipulator handles.


Drag on handles to constrain
Drag in center for all axes (based on view plane)

The move manipulator

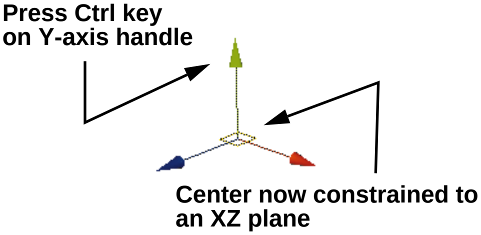
To constrain a manipulator along two axes:

- Press the **Ctrl** key and click-drag on the axis normal to the desired plane of motion.

  This now fixes the center on the desired plane, thereby letting you click-drag on the center so that you can move along the two axes. The icon at the center of the manipulator changes to reflect the new state.

To go back to a view plane focus for center:

- Press the **Ctrl** key and click on the center of the transform manipulator.


Press Ctrl key on Y-axis handle
Center now constrained to an XZ plane

Working along two axes

Note: The ability to constrain in two axes at a time is available to only the move manipulator.

Using the mouse buttons

When working with manipulators, you can use the left mouse button to select objects and interact directly with manipulators. The middle mouse button is for the active manipulator and lets you click-drag without direct manipulation.

To select objects:

- Set up selection masks.
- Click with the left mouse button.

To select multiple objects:

- Use the left mouse button and click-drag a bounding box around objects.

Or

- Press **Shift,** and with the left mouse button click on multiple objects.

To manipulate objects directly:

- Click-drag on a manipulator handle.

To manipulate objects indirectly:

- Activate a manipulator handle.
- Click-drag with the middle mouse button.

Shift gesture

The manipulators allow you to work effectively in a Perspective view panel when transforming objects.

If you want to work more quickly when changing axes for your manipulators, there are several solutions available.

To change axis focus using hotkeys:

- Press-hold on the transform keys:

  w for move;

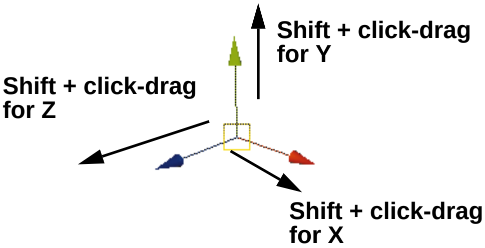
  e for rotate;

  r for scale.

- Choose an axis handle for constraining from the marking menu.

To change axis focus using shift key:

- Press the **Shift** key.
- Click-drag with the middle mouse button in the direction of the desired axis.



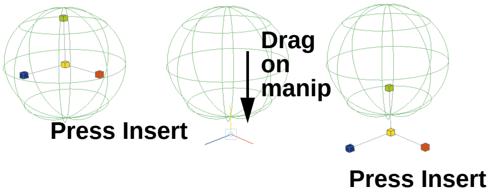
Transform manipulators

Set pivot

The ability to change the pivot location on a transform node is very important for certain types of animation.

To change your pivot point:

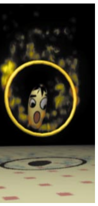
- Select one of the manipulator tools.
- Press the **Insert** key on keyboard.
- Click-drag on a manip to move pivot.
- Press **Insert** to return to manip.



Setting pivot using Insert key

Numeric input

To add values to your transformations using accurate values, you can go to the numeric input box at the right of the Feedback line. This allows you to apply exact values to the attributes associated with the current manipulator. You can use the Feedback line to check out current values and to confirm your results.



To access the feedback line

- From the **Options** menu, select **Feedback line.**

To change focus to the coordinate box:

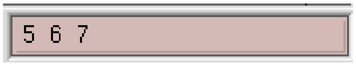
- Click on the box.

Or

- Press the **Alt** - **~** keys

To change all values at once:

- Enter three values in a row, with spaces in between.



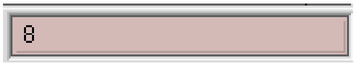
5 6 7

Numeric input field

Note: Beside the coordinate box is a button to toggle between absolute and relative values. The default is absolute.

To enter a value for the active manipulator.

- Click on the desired handle. (e.g. - Z-translate)
- Enter a single value.



8

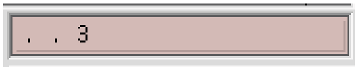
Inputting active manipulator value

Note: If no manipulator handle is active, then the single value will be applied to X.

To enter a value while preserving others:

- Type in periods (.) for channels that you want to stay the same. remember to add spaces in between.
- Enter a numeric value for the channels that will change.

  The example below would keep the X and Y values constant and change only the Z information.

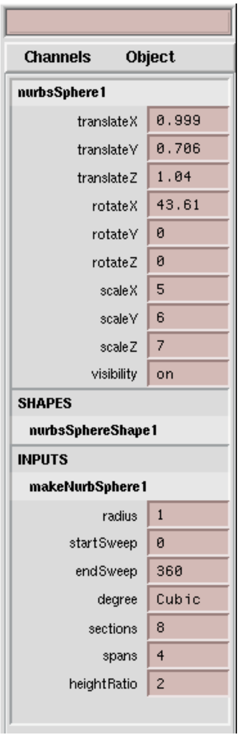


. . 3

Entering periods to keep values constant

Channel box

Another way of entering accurate values is the Channel box. This powerful panel gives you access to an object's transform node and any associated input nodes.



nurbsSphere1	
translateX	0.999
translateY	0.706
translateZ	1.04
rotateX	43.61
rotateY	0
rotateZ	0
scaleX	5
scaleY	6
scaleZ	7
visibility	on
SHAPES	
nurbsSphereShape1	
INPUTS	
makeNurbSphere1	
radius	1
startSweep	0
endSweep	360
degree	Cubic
sections	8
spans	4
heightRatio	2

Transform node

Shape node

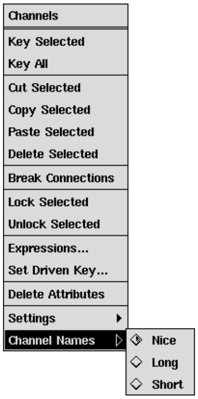
Input node

The Channel box

If you have multiple objects selected, then your changes to a channel will affect all the nodes which share that attribute.

To put one of the selected objects at the top of the Channel box so that it is visible, choose the desired node from the channel box's **Object** menu.

If you want to work with a particular channel, you can use the **Channels** menu to set keys, add expressions and complete other useful tasks. You can also change the display of the Channel box names to short MEL-based names.



Channels menu

Note: To control what channels are shown in the Channel box, you must go to the **Options** menu, and choose **General Editors** → **Channel Control**.

Channel box and manipulators

One of the features of the Channel box is the way in which you can use it to access manipulators at the transform level.

By default, the Channel box is set to show manipulators each time you tab into a new Channel box field. You will notice that as you select the channel names such as *Translate Z* or *Rotate X*, the manipulator switches from translate to rotate.

One fast way of working is to select the name of the desired channel in the Channel box, then use the middle mouse button to edit the value by click-dragging in a view panel.

There are three options for the Channel box manipulator setting:

Default manipulator setting

This setting lets you activate the appropriate field in the Channel box, and then modify the values with either the left or the middle mouse buttons.

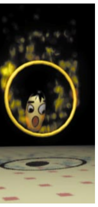
To use the default method, complete the following steps:

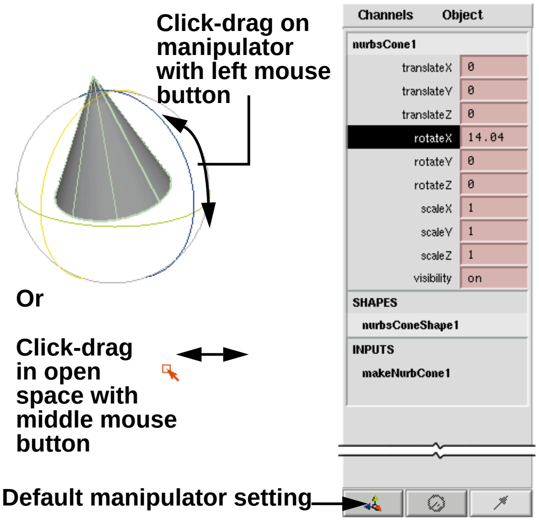
- Click on the desired channel name or within the channel's input field.
- Click-drag directly on the active manipulator with the left mouse button.

Or

- Click-drag in open space with the middle mouse button.

  This may be the easier method since you can click remotely.

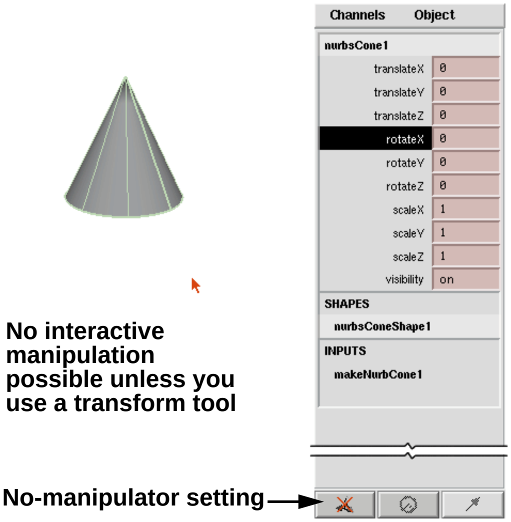



Click-drag on manipulator with left mouse button
Or
Click-drag in open space with middle mouse button
Default manipulator setting
Channels
Object

nurbsCone1	
translateX	0
translateY	0
translateZ	0
rotateX	14.04
rotateY	0
rotateZ	0
scaleX	1
scaleY	1
scaleZ	1
visibility	on

SHAPES
nurbsConeShape1
INPUTS
makeNurbCone1

Channel box default manipulator setting


No interactive manipulation possible unless you use a transform tool
No-manipulator setting
Channels
Object

nurbsCone1	
translateX	0
translateY	0
translateZ	0
rotateX	0
rotateY	0
rotateZ	0
scaleX	1
scaleY	1
scaleZ	1
visibility	on

SHAPES
nurbsConeShape1
INPUTS
makeNurbCone1

Channel box no-manipulator setting

No-manipulator setting

You can click on the manipulator icon at the bottom of the Channel box to turn manipulation off which leaves the Channel box focused on coordinate input. With this setting, you cannot use the middle or left mouse buttons for manipulation. To manipulate objects in this mode, you must do one of the following:

- Click in the channel's entry field and type the exact value.

Or

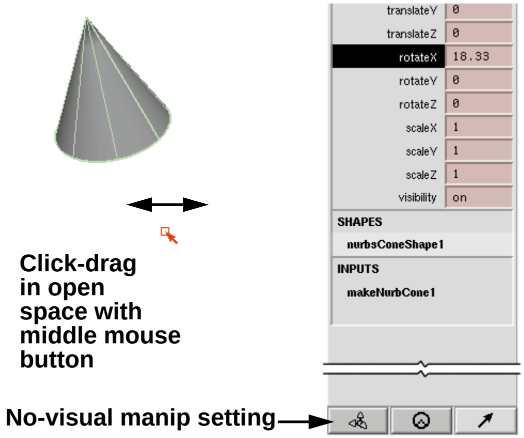
- Use one of the normal transform tools such as move, rotate and scale.

No-visual manipulator setting

A third option found on this manip button returns manipulator capability to the Channel box – but now you won't see the manipulator on the screen, as shown in the icon at the bottom of the Channel box.

- Click on the desired channel name or within the channel's input field.
- Click-drag in open space with the middle mouse button.

  You can now use the two new buttons that let you edit the speed and dropoff of the manipulations.



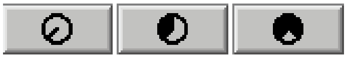
translateY	0
translateZ	0
rotateX	18.33
rotateY	0
rotateZ	0
scaleX	1
scaleY	1
scaleZ	1
visibility	on

SHAPES
nurbsConeShape1

INPUTS
makeNurbCone1

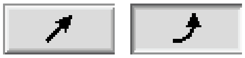
Channel box no-visual manipulator setting

The first button that becomes available with the No-visual setting is the speed button which lets you click-drag with your middle mouse button either slow, medium or fast.



Channel speed controls

The second button is the drop-off button which lets you choose between a linear motion, as you click-drag with the middle mouse button, or a click-drag that is slow at first then faster as you drag further.



Channel drop-off options

Attribute Editor

If the Channel box lets you focus on attributes that are keyable using **Set Key**, then the Attribute Editor gives you access to all the rest of the attributes/channels.

The Attribute Editor is used for all nodes in Maya. This means that shaders, textures, surfaces, lattices, render globals, etc. can all be displayed in this one type of window.

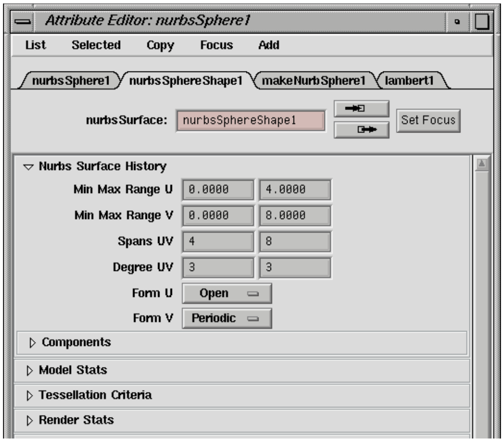
To open the Attribute Editor window:

- Select a node.
- Go to the **Window** menu and select **Attribute Editor**.

To open the Attribute Editor panel:

- Select a node.
- Go to the **Options** menu and select **Attribute Editor**. The Channel box is now replaced by an attribute editor panel.

When you open up the Attribute Editor, you get not only the active node, but also related nodes based on dependency relationships. In the example below, a sphere's transform, shape and *makeNurbSphere* nodes are all present. These are the same input and shape nodes shown in the Channel box.



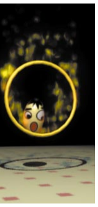
nurbsSurface: nurbsSphereShape1

Min Max Range U	0.0000	4.0000
Min Max Range V	0.0000	8.0000
Spans UV	4	8
Degree UV	3	3
Form U	Open	
Form V	Periodic	

A typical Attribute Editor

SELECTING IN MAYA

One of the most important tasks when working in Maya is your ability to select



different types of nodes and their key components.

For instance, you need to be able to select a sphere and move it, or you need to select the sphere's control vertices and move them. You also need to distinguish between different types of objects so that you can select only surfaces or only deformers.

Selection masks

To make selecting work in Maya, you have a series of selection masks available to you. This allows you to have one Select tool that is then *masked* so that it can only select certain kinds of objects and components.

The *selection mask* concept is very powerful because it allows you to create whatever combination of selecting types that you desire. Sometimes, you only want to select joints and selection handles, or maybe you want to select anything but joints. With selection masks, you get to set up and choose the select options.

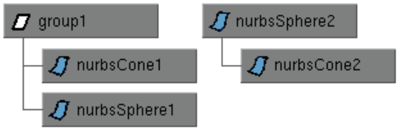
The selection user interface

The user interface for selecting offers several types of access to the selection masks. You can learn all of them now and then choose which best suits your way of working down the line.

Grouping and parenting

When working with transform nodes, you can build more complex structures by building hierarchies of these node types.

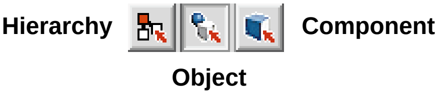
To build these structures, you can choose to *group* the nodes under a new transform node or you can *parent* one of the nodes under the other so that the lower node inherits the motion of the top node.



Grouped and parented nodes

Selection modes

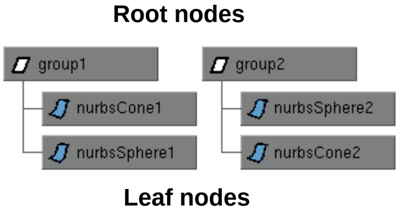
At the top of the workspace, you have several selection mask tools available. These are all organized under three main types of select modes. Each of these gives you access to either the hierarchy, object type or components.



The select modes

Scene hierarchy mode

Hierarchy mode gives you access to different parts of the scene hierarchy structure. In the example shown below, the leaf node and the root node are highlighted. This mode lets you access each of these parts of the hierarchy. You can select root nodes, leaf nodes and template nodes using the selection masks.

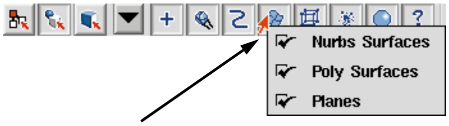


Hierarchy types

Object mode

Object mode lets you perform selections based on the object type. Selection masks are available as icons which encompass related types of objects.

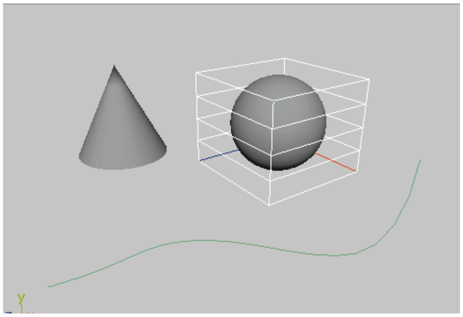
With your right mouse button, you can access more detailed options that are listed under each mask group. If you create a partial list, the mask icon is highlighted in orange.



Click with RMB on icon for list

Object mode with selection masks

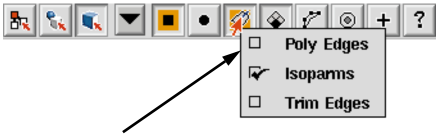
Tip: Once you choose selection masks, Maya gives priority to different object types. For instance, joints are selected before surfaces. You will need to use the **Shift** key to select these two object types together. To reset the priorities, select **Options → General Preferences** and click on the **Select** tab.



A lattice object and a curve object selected

Component mode

The shape nodes of an object contain various components such as control vertices or isoparms. To access these, you need to be in component mode.

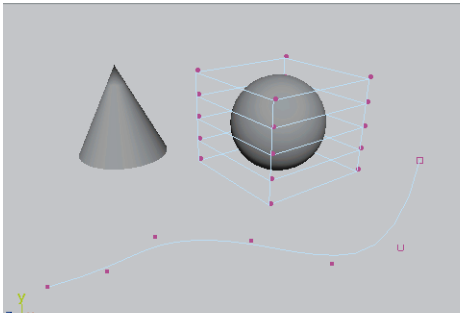


RMB list - click on icon

Component selection masks

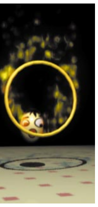
When you select an object in this mode, it first highlights the object and shows you the chosen component type; you can then select the actual component.

Once you go back to object mode, the object is selected and you can work with it. Toggling between object and component mode allows you to reshape and position objects quickly and easily.



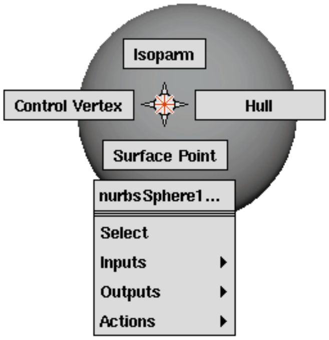
CV components and lattice point components

Tip: To toggle between object and component modes, press the **F8** key.



RMB select

Another way of accessing the components of an object is to select an object, then press the right mouse button. This brings up a marking menu that lets you choose from the various components available for that object.



The right mouse button select menu

If you then select another object, you return to your previous select mask selection. This is a very fast way of selecting components when in hierarchy mode, or for components that are not in the current select mask.

Combined select modes

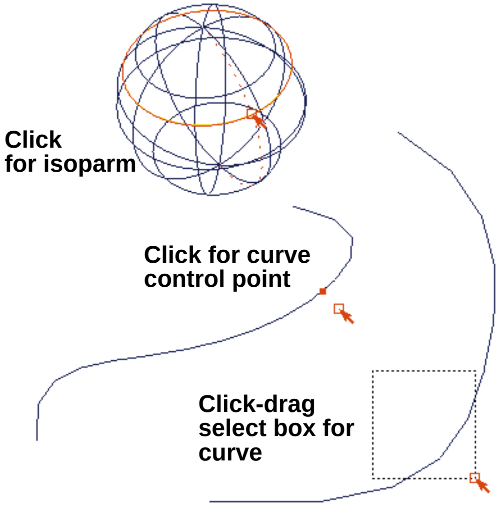
Just in front of the selection mask mode icons is a pop-up menu that gives you different preset mask options. These presets let you combine different object and component level select options.

An example would be the NURBS option. This allows you to select various NURBS-based mask types such as surfaces, curves, CVs, curve control points and isoparms.

Note: In this mode, if you want to select CVs which are not by default visible, then you must make them visible by going to the **Display** menu and selecting **NURBS Components → CVs**.

When using a combined select mode, objects and components are selected differently. Objects are selected by click-dragging a select box around a part of the object while components can be selected with direct clicking.

Note: If you have CVs shown on an object and the select box touches any of them, then you will select these components instead of the object. To select the object, you must drag the select box over part of the surface where CVs are not.


Click for isoparm
Click for curve control point
Click-drag select box for curve

NURBS select options

TOOLS AND ACTIONS

In Maya, there are a large number of menu items that let you act on your scenes in a number of ways. These menu items are broken down into two types of commands: *tools* and *actions*, each working in their own particular way.

Tip: If a menu item says "Curve tool" then it uses tool interaction. If the word "tool" is not mentioned then the menu item is an action.

Tools

Tools are designed to remain active until you have finished using them. You select a tool, use it to complete a series of steps then press the **Enter** key to complete, at which point your focus returns to your last transform tool. In most cases, the Help line at the bottom of the workspace can be used to prompt your actions when using the tool.

Earlier you were introduced to the **y** key on the QWERTY palette. By default, this button is blank because it has been left over to show the last tool used. When you pick a tool from the menus, it's icon inserts itself into the QWERTY menu.

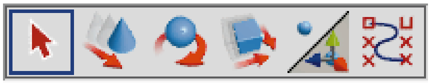
To use a tool:

- Pick and use a tool.
- Press the **Enter** key to complete.

  Your focus returns to a transform tool.

To return to the last tool used:

- Press the **y** key.



Y key with curve tool icon

Actions

Actions follow a selection-action paradigm. This means that you have to first pick something and then act on it. In Maya, this allows you to choose an action, return to editing your work, and lets you refine the results immediately.

Actions require that you have something selected before acting on it. This means that you must first find out what is required to complete the action.

To find out selection requirements of an action:

- Move your cursor over the menu item.
- Look at the Help line at the bottom left of your workspace.

  The selection requirements are displayed. For instance, a **Loft** requires *curves, isoparms* or *curves on surfaces* while **Insert Isoparm** requires that isoparms be picked.

To complete the action:

- Use either pick modes or the right mouse button pick menu to make the required selections.
- Choose the action using either the hotbox, shelf or menus.

  The action is complete and your focus returns to your last transform tool.

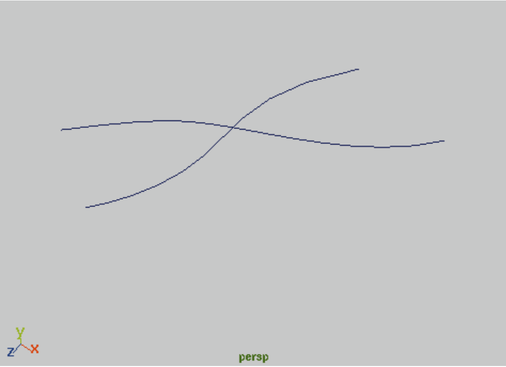
PROJECT ONE

A typical action: 2D fillet

A good example of a typical action is a 2D fillet. As with all actions, you must start with an understanding of what the tool needs before beginning to execute the action.

1 Draw two curves

- Select **Create → CV Curve Tool**.
- Place several points for one curve.
- Press **Enter** to complete.
- Press the **y** key to refocus on curve tool.
- Draw the second curve so that it crosses the first.
- Press the **Enter** key to complete



Two curves for filleting

2 Find out 2D fillet requirements

- Move your cursor over the **Modeling → Edit Curves → Curve Fillet** menu item.
- Look in the Help line to find out what kind of pick is required.

  The Help line is asking for *curve parameter points*.

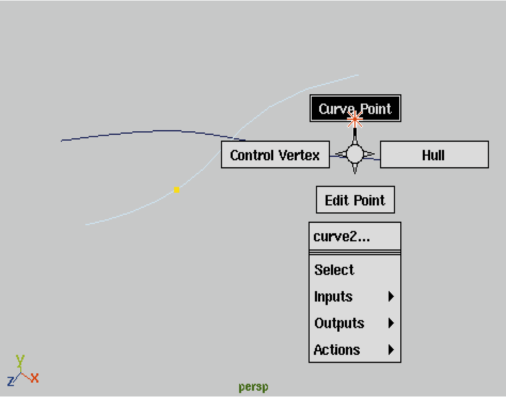
3 Pick the first curve point

- Click on the first curve with the right mouse button.
- Pick **Curve Point** from the selection marking menu.
- Click on the curve to place the point on the side you want to keep.

4 Pick the second curve point

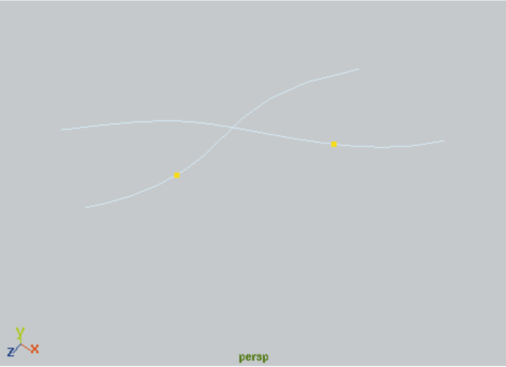
- Click on the second curve with the right mouse button.
- Pick **Curve Parameter point** from the selection marking menu.
- Press the **Shift** key and click on the curve to place the point on the side of the curve you want to keep.

The **Shift** key lets you add a second point to the selection list without losing the first curve point.



RMB pick of curve parameter point

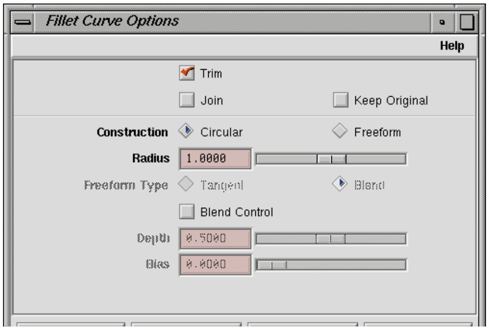
Note: You must first use the marking menu then use the **Shift** key to add a second point to the selection list, otherwise the selection menu will not appear.



Two curve points in place

5 Fillet the curves

- Select **Modeling** → **Edit Curves** → **Curve Fillet - ❑** to open the tool options.
- Turn the trim option on.
- Click on the **Fillet** button.



Fillet Curve Options

Help

☑ Trim

□ Join □ Keep Original

Construction ◆ Circular ◇ Freeform

Radius 1.0000

Freeform Type ◇ Tangent ◆ Blend

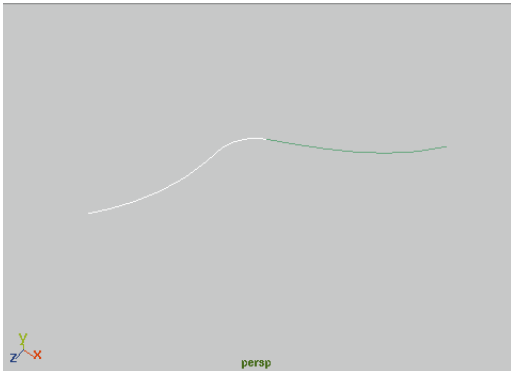
□ Blend Control

Depth 0.5000

Bias 0.0000

Fillet tool options window

- Close the Fillet Curve Options window.



Final filleted curves

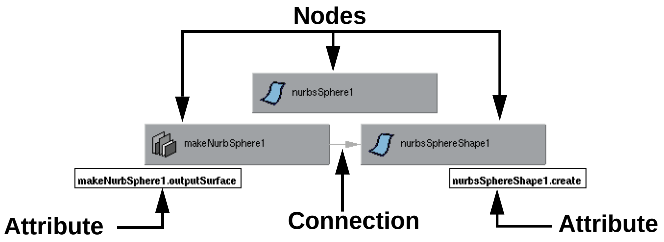
Conclusion

You now know how to navigate the Maya user interface and how the tools and actions work. The skills you have learned here will be applied throughout the rest of this book. It is now up to you to choose how you want to use the interface. Try out the different techniques taught here as you work through the Learning Maya projects.

In the next chapter, you will explore Maya's Dependency graph. You will learn about the different Maya nodes and how to build them into hierarchies and procedural animations.

6 The Dependency Graph

In the first four lessons of this book, you were able to interactively animate a ball bouncing. Along the way, you encountered many nodes which helped you build up and animate the scene. You came across input nodes, hierarchy nodes, shading group nodes and texture nodes, as well as emitter and particle nodes. These nodes represent key elements within Maya – each node containing important attributes which help you define and animate your scenes.

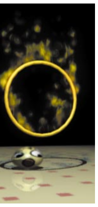


Nodes, attributes and connections

Maya's architecture is defined by this node-based system that is known as the *Dependency graph*. Each of your nodes contains attributes which can be connected to other nodes. If you wanted to reduce Maya to its bare essentials, you could basically describe it as *nodes with attributes that are connected*. This node-based approach gives Maya its open and flexible procedural characteristics.

In this chapter, you are going to explore nodes, attributes and connections by animated objects at various levels. You will explore how attributes are connected by Maya and how you can connect them yourself. You will also learn how to distinguish scene hierarchies from object dependencies.

This chapter will at first seem a bit abstract, but in the end you will be able to see how the various nodes contribute to an animated scene which will help you in later lessons.



Hierarchies and dependencies

If you understand the idea of *nodes with attributes that are connected*, then you will understand the Dependency graph. By building a simple primitive sphere, you can see what this means in Maya.

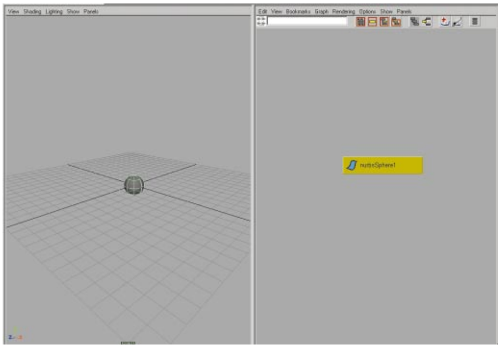
1 Set up your view panels

To view nodes and connections in a diagrammatic format, the Hypergraph panel is required along with a Perspective view.

- Select **Panels → Layouts → 2 Side by Side**.
- Set up a Perspective view in the first panel and a Hypergraph view in the second panel.
- Dolly into the Perspective view to get closer to the grid.

2 Create a primitive sphere

- Select **Create → NURBS Primitives → Sphere**.
- Press **5** to turn on smooth shading and **3** to increase the surface smoothness of the sphere.



New sphere

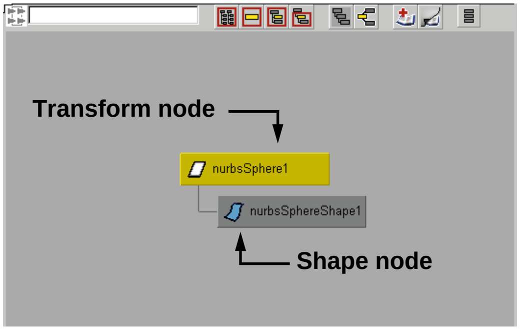
3 View the shape node

In the Hypergraph panel, you are currently looking at the scene view. The scene view is focused on *transform nodes*. This type of node lets you set the position and orientation of your objects.

Right now, only a lone *nurbsSphere* node is visible. In actual fact, there are two nodes in this hierarchy but the second is hidden by default. At the bottom of most hierarchies, you will find a *shape node* which contains the information about the object itself.

- In the Hypergraph, select **Options → Display → Shape nodes**.

  You can now see the *transform node* which is in effect the positioning node, and the *shape node* which contains information about the actual surface of the sphere. The transform node defines the position of the shape below.

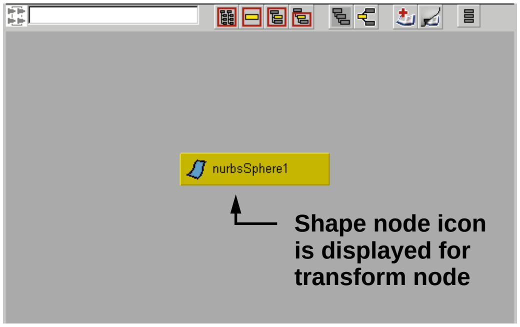


Transform and shape nodes

- In the Hypergraph panel, select **Options → Display → Shape nodes** to turn these off.

  Notice that when these nodes are expanded, the shape node and the transform node have different icons.

When collapsed, the transform node takes on the shape node's icon to help you understand what is going on underneath.

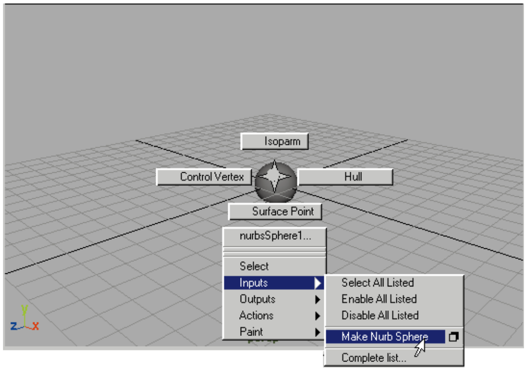


Transform node on its own

4 View the dependencies

To view the dependencies that exist with a primitive sphere, you need to take a look at the up and downstream connections.

- Click on the sphere with the right mouse button and select **Inputs → Make Nurb Sphere** from the marking menu.



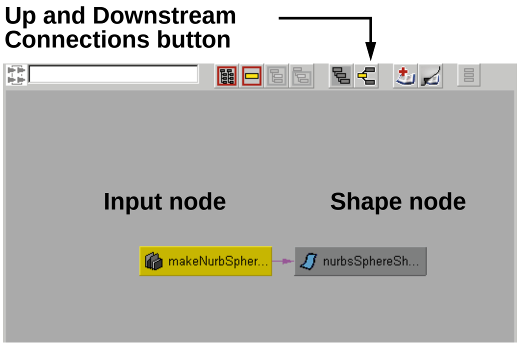
Selection marking menu

Note: You can also select the input node by selecting on it in the Channel box.

- In the Hypergraph panel, click on the **Up and Downstream Connections** button.

The original transform node is now separated from the shape node. While the transform node has a hierarchical relationship to the shape node, their attributes are not dependent on each other.

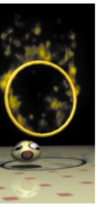
The *input node* called *makeNurbSphere* is a result of the original creation of the sphere. The options set in the sphere tool's option window, have been placed into a node that feeds into the shape node. The shape node is dependent on the input node. If you change values in the input node, then the shape of the sphere changes.



Sphere dependencies

5 Edit attributes in the Channel box

In the Channel box, you can edit attributes belonging to the various nodes. All of the node types can be found in the Channel



box. This lets you affect both hierarchical relationships and dependencies.

If you edit an attribute belonging to the *makeNurbSphere* node, then the shape of the sphere will be affected. If you change an attribute belonging to the *nurbSphere* transform node, then the positioning will be changed. Use the Channel box to help you work with the nodes.

- For the transform node, change the **Rotate Y** value to **45**.
- For the *makeNurbSphere* input node, change the **Radius** to **3**.

  You can set attribute values to affect either the scene hierarchy or the Dependency graph.

Shading group nodes

In earlier lessons, the word *node* was used a great deal when working with shading groups. In fact, shading group nodes create dependency networks which work the same way as shape nodes.

1 Create a shading group

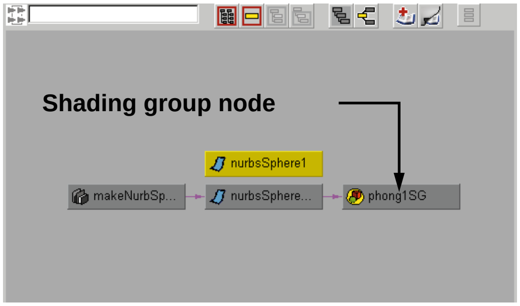
When you create a shading group you are creating two main nodes that are connected – the shading group node and the material node.

- Select **Window → Hypershade**.
- Select **Create → Materials → Phong**. In the perspective view panel click on the sphere to assign the material.
- Select the sphere in the perspective panel then click on the **Up and Downstream Connections** button.

In the Hypergraph view, notice how the input node is connected to the shape node which relates to the phong shading group.

A line is now drawn between the sphere's shape node and shading group node. This is because the shading group is dependent on the surface in order to render.

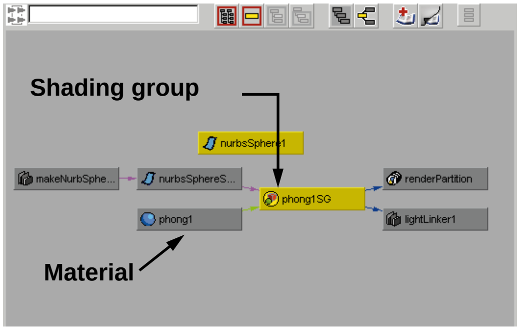
Every time you assign a shading group to an object you make a dependency graph connection.



Shading group dependencies

- Select the *nurbSphere* node and the *phongSG* node in the Hypergraph.
- Again, click on the **Up and Downstream Connections** button.

You can now see how the *phong* material node and the sphere's shape node both feed the shading group. You can move your cursor over any of the connecting lines to see the attributes that are being connected.



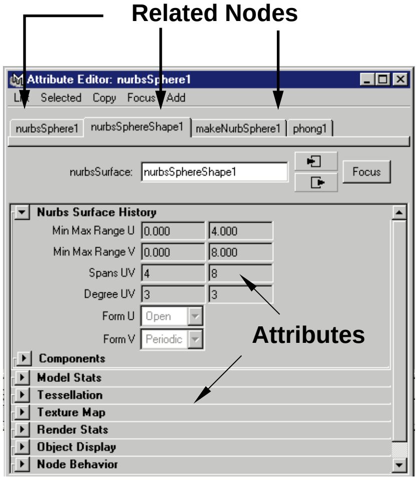
Assigned shading group

2 Open the Attribute Editor

You have seen how the nodes in the Hypergraph and the Channel box have been used to view and edit attributes on connected nodes. Now you will see how the Attribute Editor displays nodes, attributes and connections.

- Click on the **Scene Hierarchy** button in the Hypergraph Panel to go back to a scene view.
- Select the sphere's transform node.
- Press **Ctrl a** to open the Attribute Editor.

  In this important window, you will see several tabs each containing groups of attributes. Each tab represents a different node. All the tabs displayed represent parts of the selected node's Dependency graph that are related to the chosen node. By bringing up several connected nodes, you have easier access to particular parts of the graph.



nurbsSurface: nurbsSphereShape1

Min Max Range U 0.000 4.000
Min Max Range V 0.000 8.000
Spans UV 4 8
Degree UV 3 3
Form U Open
Form V Periodic

Nodes and attributes in Attribute Editor

- Close this window.

Note: In Maya, the Dependency graph lets you focus on one part of the graph at a time.

Making your own connections

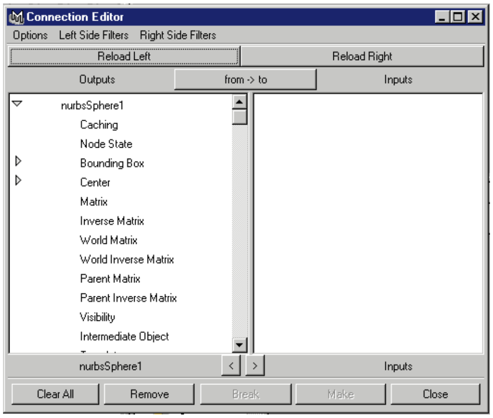
To help you understand exactly what a Dependency graph connection is, you are going to make your own connection and see how it affects the graph.

1 Open the Connection Editor

- Select **Window** → **General Editors** → **Connection Editor...**
- Click on the **Reload Left** button.

  The selected transform node is loaded into the left column. All of the attributes belonging to this node are listed.

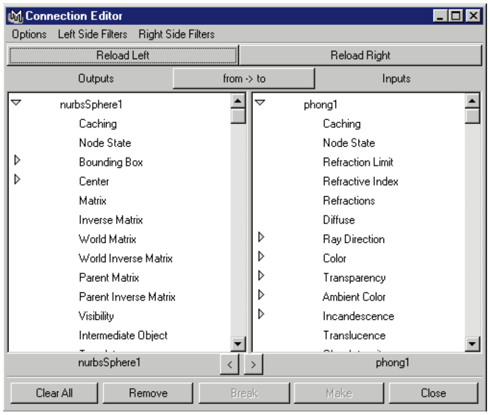
Note: There are more nodes here than you saw earlier in the Channel box. The Channel box only shows attributes that have been set as keyable. Other nodes can be found in the Attribute Editor.



Transform node in Connection Editor

2 Add phong as the output node

- In the Hypergraph, select **Rendering → Show Materials**.
- Select the *phong1* material node.
- In the Connection Editor, click on the **Reload Right** button.

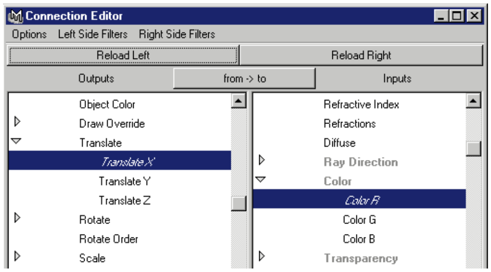


Material node in Connection Editor

3 Make connections

You will now connect some attributes in from the transform node to the material node.

- In the left hand column, scroll down until you find the *Translate* attributes.
- Click on the arrow to open this attribute type and see the *Translate X, Y* and *Z* attributes. Be sure not to click on the name.
- In the right hand column, scroll down until you find the *Color* attribute.
- Click on the arrow to open this attribute type and see the *Color R, G* and *B* attributes.
- Click on the **Translate X** attribute in the left hand column.
- Click on the **Color R** in the right hand column.



Connected attributes

- Use the same method to connect:

  Translate Y to **Color G**;

  Translate Z to **Color B**.

  In the modeling view, the sphere is still showing gray because hardware shading cannot display this kind of connection.

4 Test render the view

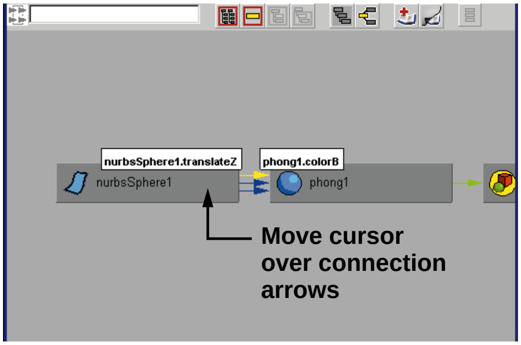
- Go to the **Rendering** menu set.
- Select **Lights** → **Directional Light**.
- Select **Render** → **Render into New Window**.

  The ball is rendered as black against a black background. Since the sphere's XYZ, coordinates are set to 0, 0, 0, the colors RGB are the same. An RGB setting of 0, 0, 0 gives you black.

5 View the connections

- In the Hypergraph panel, select the *phong* node then click on the **Up and Downstream Connections** button.
- Move your cursor over one of the arrow connections between the transform node and the material node.

  The arrow is highlighted and the connected attributes are displayed. You now see the diagrammatic results of your action. Now you should see the effect in the Perspective view.



Move cursor over connection arrows

Viewing attribute connections

6 Move the sphere

- In the Perspective view, **select** the sphere.
- **Move** the sphere along the X-axis.
- Select **Render** → **Render into New Window**.

  The color of the sphere is now red. By increasing the value of the translation along X, you add red to the color.

  Of course, once you pass a value of 1 for the translation, the color value gets too high. This example works best if you only move the sphere up to 1 unit in each direction.
- Try moving the sphere along each of the three main axes to see the colors change.

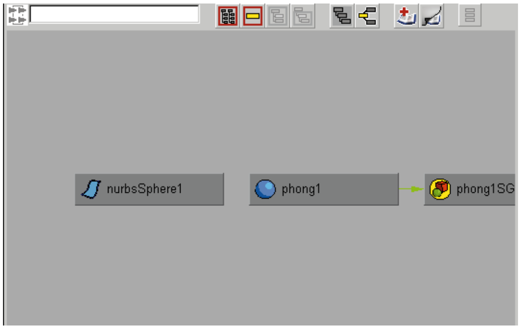
Adding a texture node

While it is a fun and educational exercise to see the material node's color dependent on the position of the ball, it may not be very realistic. You will now break the existing connections and map a texture node in their place.

1 Delete connections

You can delete the connections in the Hypergraph view.

- In the Hypergraph view panel, select one of the three connection arrows between the transform node and the material node.
- Press the **Backspace** key to delete the connection.
- Repeat for the other two connections between these nodes.



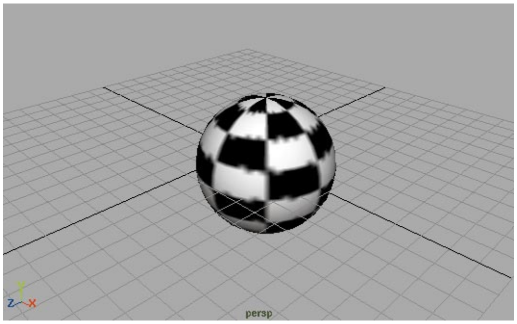
Broken connections

2 Add a checker texture map

You will now use the Attribute Editor to help you add a texture to the existing shading group.

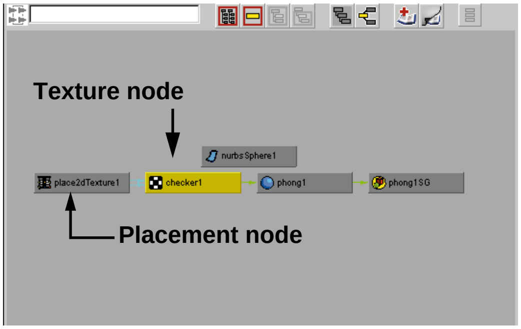
- Click on the *phong1* material node.
- Press **Ctrl a** to open the Attribute Editor.
- Click on the **Map** button next to Color.
- Choose a *Checker* texture from the Create Render node window.
- If you have hardware texturing, move your cursor over the Perspective view panel and press **6**.

Tip: If you do not have hardware texturing, then you may want to use a wireframe view – by pressing **4** – so that you can see the object's rotation being animated.



Textured sphere

In the Hypergraph, you can see the dependencies building up for the shading group. The texture is built using two nodes. The checker node which contains the procedural texture attributes, and the placement node which contains attributes that define the placement of the texture on the assigned surfaces.



Shading group network

Animating the sphere

When you animate in Maya, you are changing the value of an attribute over time. Using keys, you set these values at important points in time, then use tangent properties to determine how the attribute value changes in between the keys.

The key and tangent information is placed in a separate animation curve node that is then connected to the animated attribute.

1 Select the sphere

- In the Hypergraph panel, click on the **Scene Hierarchy** button.
- **Select** the *nurbsSphere* transform node.

2 Return the sphere to the origin

Since you earlier moved the sphere along the three axes, it's a good time to set it back to the origin.

- In the Channel box, set the **Translate X**, **Y** and **Z** to **0, 0, 0**.
- In the Channel box, change the **Rotate Y** attribute to **0**.

3 Animate the sphere's rotation

- In the Time slider, set the playback range to **120** frames.
- In the Time slider, go to frame **1**.
- Click on the **Rotate Y** channel name in the Channel box.
- Click with your right mouse button and select **Key Selected** from the pop-up menu.

  This sets a key at the chosen time.
- In the Time slider, go to frame **120**.
- In the Channel box, change the **Rotate Y** attribute to **720**.
- Click with your right mouse button and select **Key selected** from the pop-up menu.
- Playback the results.

  The sphere is now spinning.

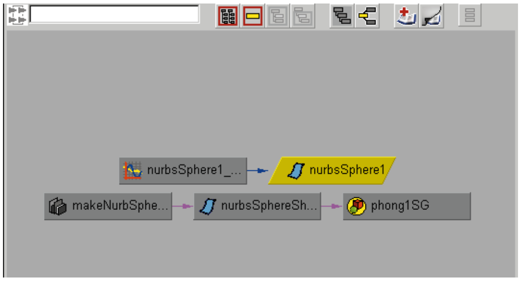
4 View the dependencies

- In the Hypergraph panel, click on the **Up and Downstream Connections** button.

  You see that an animation curve node has been created and then connected to the transform node. The transform node is now shown as a trapezoid to indicate that it is now connected to the animation curve node. If you click on the connection arrow, you will see that the connection is to *Rotate Y*.

  If you select the animation curve node and open the Attribute Editor, you will see that each key has been recorded along with value, time and tangent information. You can

actually edit this information here, or use the Graph Editor where you get more visual feedback.



Connected animation curve node

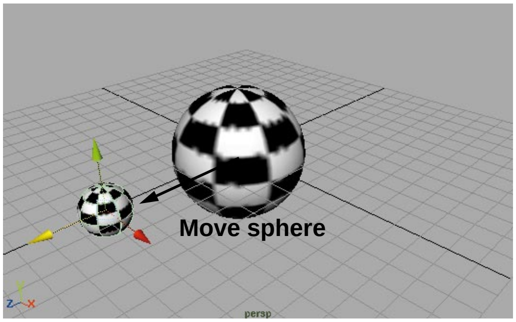
Building scene hierarchies

So far, you have worked a lot with the dependency connections but not with the scene hierarchy. In a hierarchy, you always work with transform nodes. You can make one transform node the *parent* of another node, thereby creating a child which must follow the parent.

You will build a hierarchy of spheres that are rotating like planets around the sun. This example is a helpful way to understand how scene hierarchies work.

1 Create a new sphere

- Go to frame **1**.
- In the Hypergraph panel, click on the **Scene Hierarchy** button.
- Select **Create** → **NURBS Primitives** → **Sphere**.
- **Move** the sphere along the Z axis until it sits in front of the first sphere.
- Press **3** to increase the display smoothness of the sphere.
- Go to the **Rendering** menu set.
- Select **Shading** → **Assign Shading Group** → **phongSG**.



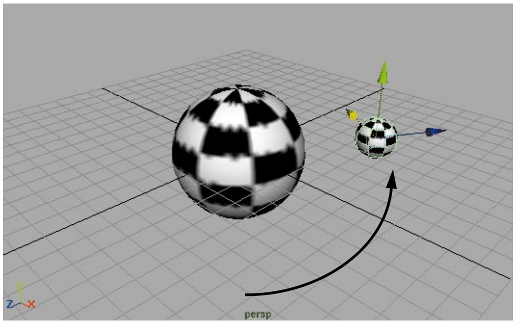
Second sphere

2 Parent the sphere

- Press the **Shift** key and select the first sphere.
- Press the **p** key to parent the new sphere to the first.

  The **p** key does the same thing as selecting **Edit** → **Parent**.
- Play back the scene.

  The second sphere rotates along with the first sphere. It has inherited the motion of the original sphere.



Rotating hierarchy

3 Animate the new sphere's rotation

While child nodes in a hierarchy inherit the motion of the parent, they can also contain their own animation.

- Go to frame **1**.
- With the new sphere selected, go to the Channel box and select **Key Selected** for the **Rotate Y** channel.
- Go to frame **120**.
- In the Channel box, change the **Rotate Y** attribute to **-1440**.
- **Key Selected** for the **Rotate Y** channel.
- Playback the scene.

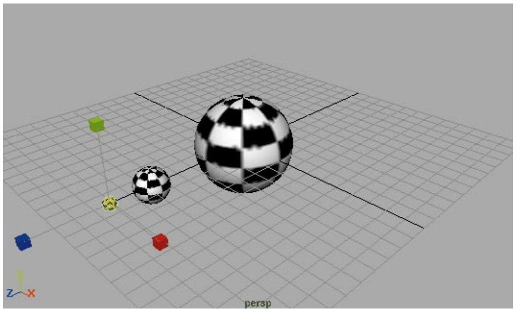
  The second sphere rotates in the opposite direction as it revolves around the first sphere.

Note: One way of thinking of the hierarchy is to think of a person walking on our own planet. As the planet revolves around its axis, the person revolves too. The person can also walk, jump or spin.

4 Create another sphere

To make this point more clear, you will add a third sphere to the hierarchy to see what happens.

- Go to frame 1.
- Select **Create → NURBS Primitives → Sphere**.
- **Move** the sphere along the Z-axis until it sits in front of the first sphere.
- **Scale** the sphere in all three axes to about one third its size.
- Press **3** to increase the display smoothness of the sphere.
- Go to the **Rendering** menu set.
- Select **Shading → Assign Shading Group → phong1Grp**.

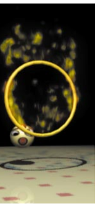


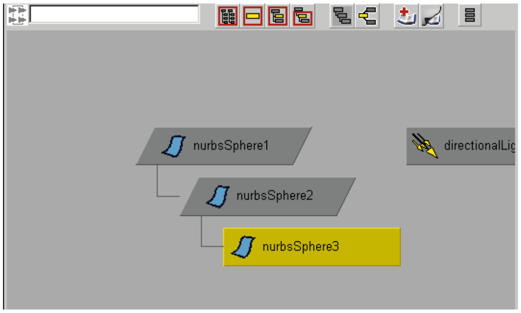
Third sphere scaled down

5 Parent the sphere

- In the Hypergraph panel, click on the **scene hierarchy** button.
- Now use the middle mouse button to click-drag the new transform node onto the second sphere's transform node. This is another method for parenting nodes.
- Play back the scene.

  The third sphere revolves with the rotation of the second sphere. This hierarchy is then revolved around the first sphere.





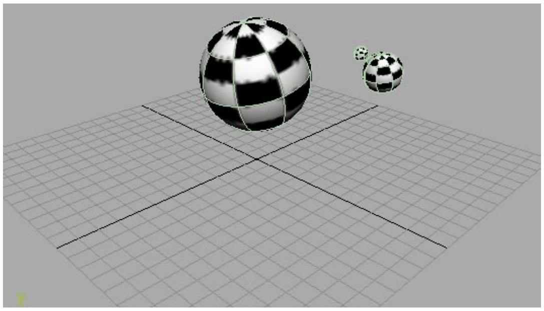
Scene hierarchy view

6 Move and rotate the hierarchy

- In the Perspective view panel, **select** the first sphere.
- Select **Edit** → **Group** to add another node to the hierarchy.

  This node is now the root of the hierarchy. You can use this to position the nodes below.
- **Move** the hierarchy up along the Y-axis about 5 units.
- **Rotate** the hierarchy around the X-axis about 15 degrees.
- Play back the results.

  The hierarchy is now rotating at an angle. You have changed the axis of the whole system.



Repositioned hierarchy

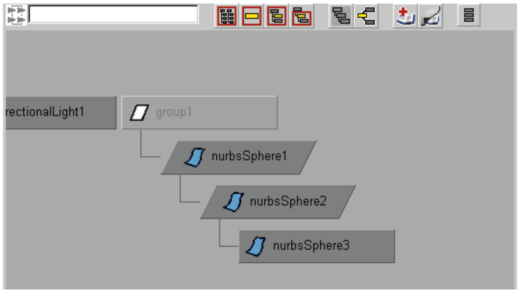
Understanding how hierarchies work will be an important part of working with Maya. In upcoming lessons, keep an eye on how hierarchies are built, especially when you begin building characters.

Hiding objects

Before moving onto a more complex animation, you will hide the existing hierarchy. This will let you focus on the second part of this lesson.

1 Hide the grouped hierarchy

- With the new group selected, select **Display** → **Hide** → **Hide Selection**.



Invisible node

Procedural animation

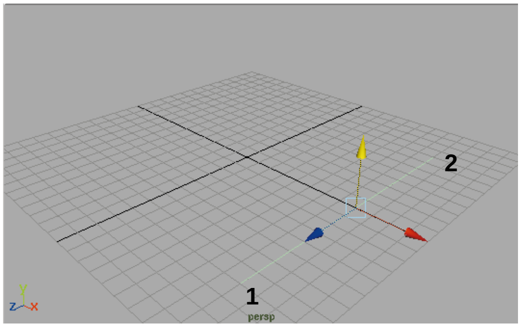
If Maya's procedural nature is defined as *nodes with attributes that are connected*, then a procedural animation would be set up by animating attributes at various levels of a Dependency graph network.

You will now build a series of animated events that build on each other to create the final result.

1 Create an edit point curve

- Select **Create** → **EP Curve Tool**.
- Press the **x** key to turn on grid snap.

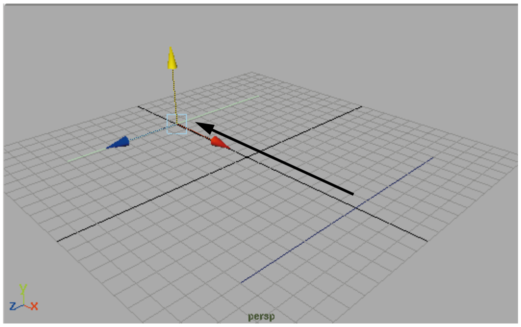
- Draw a curve as shown below.
- When you are finished, press **Enter** then select **Modify** → **Center Pivot**.



New curve

2 Duplicate the curve

- Select **Edit** → **Duplicate**.
- **Move** the new curve to the opposite end of the grid.

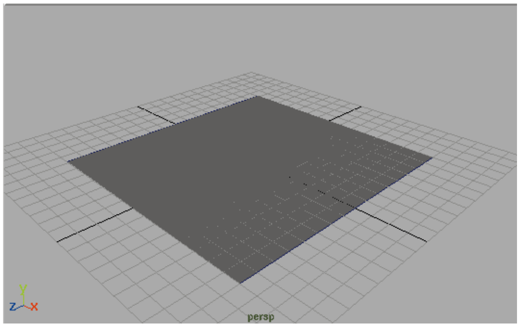


Moved curve

3 Create a lofted surface

A lofted surface can be created using two or more profile curves.

- Click-drag a selection box around both of the curves.
- Select **Surfaces** → **Loft**.
- Press **3** to increase the surface display smoothness.



Lofted surface

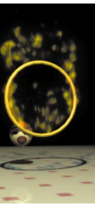
4 Change your panel display

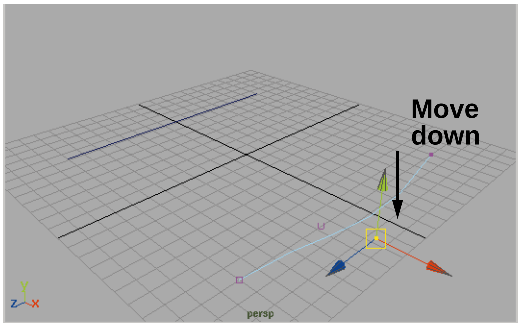
- In the Hypergraph panel, select **Panels** → **Perspective** → **persp**.
- In the Hypergraph panel, select **Show** → **None** then **Show** → **Curves.**

Now you have two Perspective views. One shows the surface in shaded mode and the second shows only the curves. This makes it easier to pick and edit the curves in isolation from the surface itself.

5 Edit CVs on the original curves

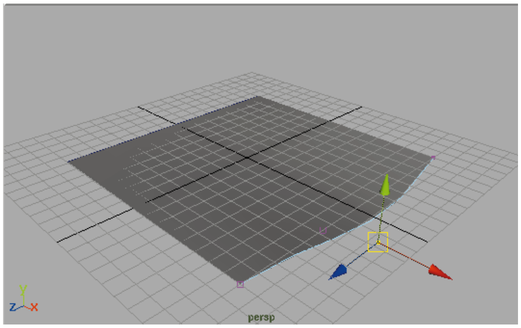
- **Select** the first curve.
- Click with your right mouse button to bring up the selection marking menu and select **Control Vertex**.
- Click-drag a pick box over one of the CVs and **Move** it down.





Edited profile curve

In the original Perspective view, you can see the effect on the lofted surface. Since the surface was dependent on the shape of the curve, you again took advantage of the Dependency graph.



Resulting surface update

Note: The dependencies associated with models are sometimes referred to as construction history. By updating the input shape, you have in effect updated the history of the lofted surface.

Creating a curve on surface

You will now build a curve directly onto the surface. This curve will become dependent on the shape of the surface for its own shape.

The surface was built as a grid of surface lines called *isoparms*. These lines help define a separate coordinate system specific to each surface. Whereas world space coordinates are defined by X, Y and Z, surface coordinates are defined by U and V.

1 Make the surface live

So far, you have drawn curves into the world space coordinate system. You can also make any surface in Maya into a *live* surface and draw into the UV space of the surface.

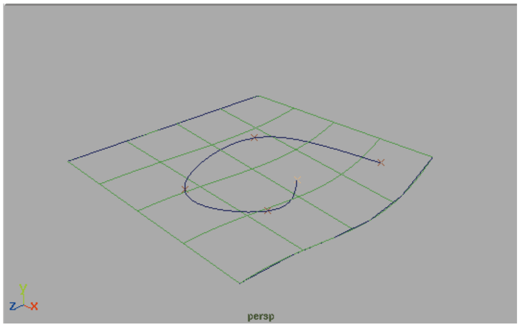
- **Select** the lofted surface.

  The CVs on the curve disappear and you are able to focus on the surface.

- Select **Modify** → **Make Live**.
- Select **Display** → **Grid** to turn off the ground grid.

2 Draw a curve on the surface

- Select **Create** → **EP Curve Tool**.
- Draw a curve on the live surface.



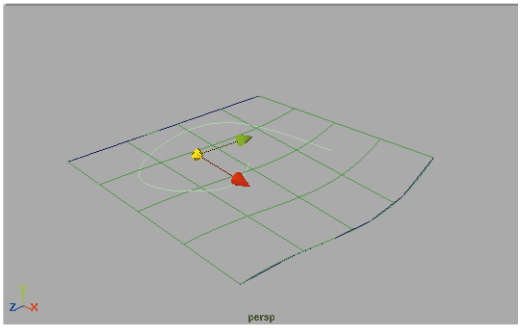
New curve on surface

3 Move the curve on surface

- Press the **Enter** key to complete the curve.
- Select the **Move** tool.

The move manipulator looks a little different this time. Rather than three manipulator handles, there are only two. One is for the U direction of the surface and the other is for the V direction.

- Click-drag on the manipulator handles to move the curve around the surface space.



Moving the curve on surface

Tip: This UV space is the same used by texture maps when using 2D placement nodes.

4 Make the ground grid live

- Click in empty space to deselect the curve on surface.
- Select **Modify → Make Live**.

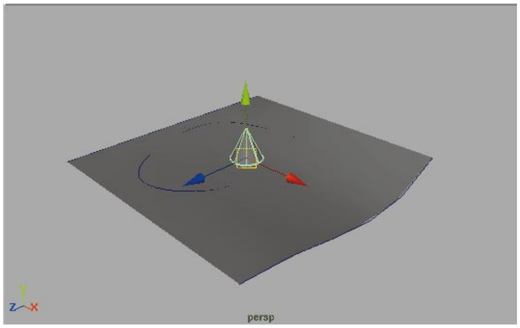
With nothing selected, the ground grid becomes the active or live grid.

Create group hierarchy

You are now going to build another hierarchy. This time you will group two primitives, then animate the group along the curve on surface using path animation.

1 Create a primitive cone

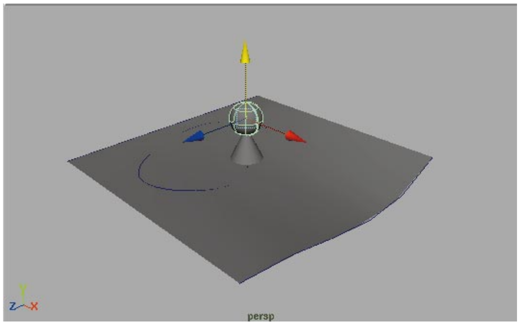
- Select **Create → NURBS Primitives → Cone**.
- Press the **3** key to increase its display smoothness.



New primitive cone

2 Create a primitive sphere

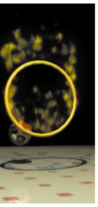
- Select **Create → NURBS Primitives → Sphere**.
- Press the **3** key to increase its display smoothness.
- **Move** the sphere above the cone.



Second primitive object

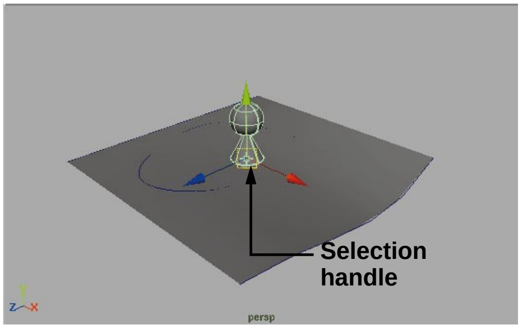
3 Group the two objects

- **Select** the cone and the sphere.
- Select **Edit → Group**.



- Select **Display** → **Object Components** → **Selection Handles**.

  The selection handle is a special marker that will make it easier to pick the group in object selection mode.



Grouped objects with selection handle

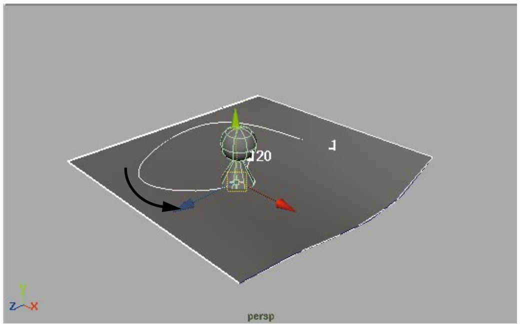
Create a path animation

To animate the new group, you will attach it to the curve on surface. You can use the curve on surface to define the group's position over time.

1 Attach to the curve on surface

- With the group still selected, press the **Shift** key and select the curve on surface.
- Go to the **Animation** menu set.
- Select **Animate** → **Paths** → **Attach to Path** - ❐. In the option window, make sure that the **Follow** option is turned **Off**.
- Click **Attach** then **Close**.
- Play back the results.

As the group moves along the path curve, you will notice that it is always standing straight up.



Path animation

2 Constrain to the surface normal

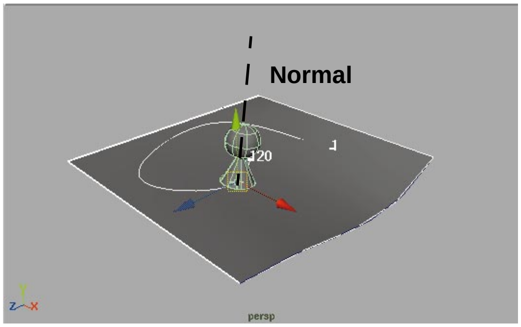
You will now constrain the orientation of the group to the normal direction of the lofted surface. The normal is like the third dimension of the surface's UV space.

- Click in open space to deselect the active objects.
- **Select** the loft surface.
- Press the **Shift** key and select the grouped primitives using the selection handle.
- Select **Constrain** → **Normal** - ❐. In the option window, set the following:

  Aim Vector to **0**, **-1**, **0**;

  Up Vector to **1**, **0**, **0**.
- Click **Add** then **Close**.
- Play back the results.

Now the group is orienting itself based on the normal direction of the surface. The group is now dependent on the surface in two ways. Firstly, its position is dependent on the path curve, which is dependent on the surface for its shape. Secondly, its orientation is directly dependent on the surface's shape.



Constrained orientation

Layer the animation

In Maya, the various parts of the Dependency graph can all be animated to create exciting results. To see the Dependency graph in motion, you will animate different nodes within the network to see how the dependencies react.

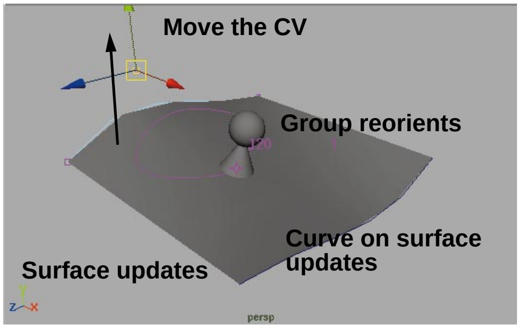
1 Edit the loft curve shape

Since the shape of the surface is dependent on the original loft curves, you will start by animating the shape of the second curve.

- **Select** the second loft curve. You may want to use the second Perspective panel which is only displaying curves.
- Click with your right mouse button to bring up the selection marking menu and select **Control Vertex**.

  Control vertices help define the shape of the curve. By editing these, you are editing the curve's shape node.
- Click-drag a pick box over one of the CVs and **Move** it up to a new position.

As you move the CV, the surface updates its shape, which in turn redefines the curve on surface and the orientation of the group. All the dependencies are being updated.

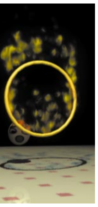


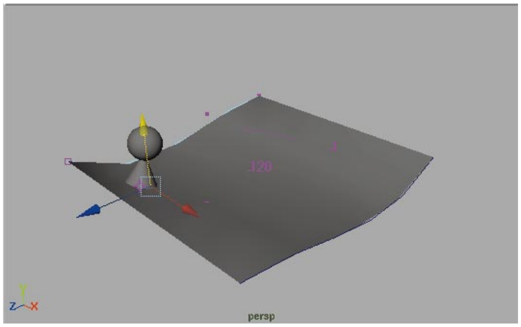
Updating the dependencies

2 Set keys on the CV position

- Go to frame **1** in the Time slider.
- Press **s** to set key.
- Go to frame **120** in the Time slider.
- Press **s** to set key.
- Go to frame **60** in the Time slider.
- **Move** the CV down to a new position.
- Press **s** to set key.
- Play back the results.

  You can see how the dependency updates are maintained as the CV is animated. You are animating the construction history of the lofted surface and the connected path animation.





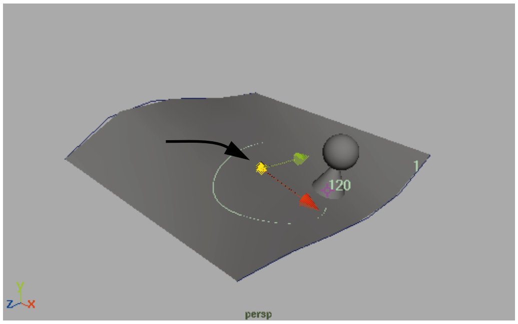
Animated history

3 Animate the curve on surface

To add another layer of animation, you will key the position of the curve on surface.

- **Select** the curve on surface.
- Go to frame **1** in the Time slider.
- Press **s** to set key.
- Go to frame **120** in the Time slider.
- **Move** the curve on surface to another position on the lofted surface.

  Press **s** to set key.



Animated curve on surface

4 Assign the phong shading group

To make it easier to see the animating objects, apply the checker shading group to the primitive group. You will also make visible the animated hierarchy to see all the pieces together.

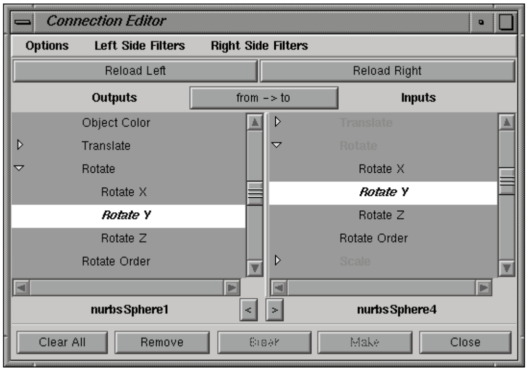
- **Select** the primitive group using its selection handle.
- Go to the **Render** menu set.
- Select **Shading** → **Assign Shading Group** → **phong1SG**.
- Select **Display** → **Show** → **Show Last Hidden**.
- Play back the scene.

5 Connect two of the spheres

To add your own connection into the equation, you will now connect the rotation of the large sphere from the original hierarchy with the sphere that sits on the cone.

- Select **Window** → **General Editors** → **Connection Editor**.
- **Select** the large sphere from the original hierarchy.
- In the Connection Editor, click on the **Reload left** button.
- Select the sphere that is sitting on top of the cone *without* using the selection handle.
- In the Connection Editor, click on the **Reload right** button.
- In the left hand column, scroll down to the **Rotate** section.
- Click on the arrow next to **Rotate** to open this section then click on **Rotate Y**.

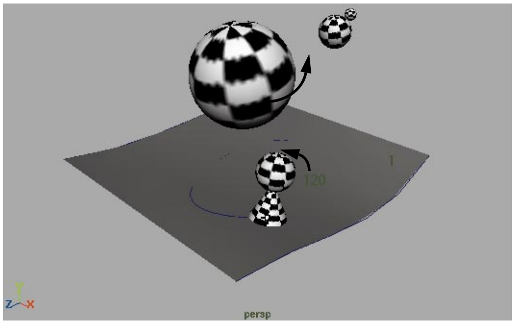
- Repeat these steps for the right hand column. When you click on the **Rotate Y** attribute for the second sphere, the two are connected.



Connected attributes

- Play back the scene.

Now the sphere on top of the cone animates in relation to the larger sphere.



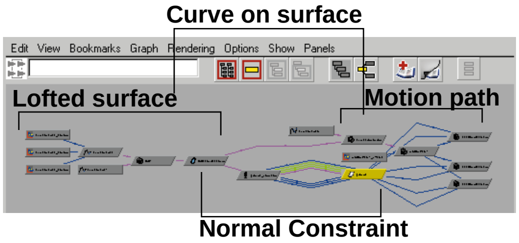
Animated scene

6 View the dependencies

Of course, you can view the dependency network that results from all these connections in the Hypergraph view, which will probably be a bit more complex than anything you have seen so far.

- **Select** the primitive group that is attached to the motion path.
- Open the Hypergraph panel and click on the **Up and Downstream Connections** button.

The resulting network contains the various dependencies that you built during this example.



The dependency network

7 Save your work

Conclusion

Maya's procedural qualities are tied to how the Dependency graph uses nodes, attributes and connections. You can see how deep these connections can go and how they are maintained throughout the animation process. Similar techniques can be used on other node types throughout Maya.

Obviously, you don't have to use the Hypergraph and the Connection Editor to build, animate and texture map your objects. But in most cases, you will be thinking more about the motion of your character's walk or the color of their cheeks. In this way, it is a good idea to know that the Dependency graph underlies everything that you do and can always be used to your advantage.

Lesson 7

Modeling

Lesson 8

Organic Modeling

Lesson 9

Animation

Lesson 10

Secondary Animation

Lesson 11

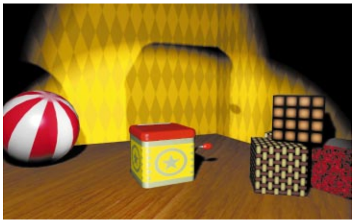
Rendering

Project Two

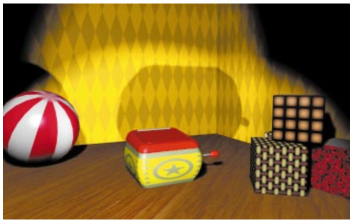
PROJECT TWO

In this project, you will animate a Jack-in-the-box. You will begin by modeling the box. You will then use CV pulling and Maya Artisan tools to create Jack's face and head.

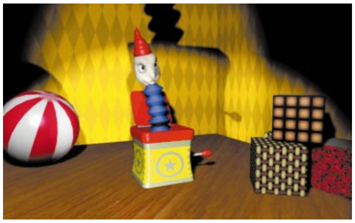
To animate the whole scene, you will set keyframes on the rotation of the box's lid and crank then you will use inverse kinematics to animate Jack emerging from the box. Deformations will then be added for some squash and stretch. This project includes the following lessons:



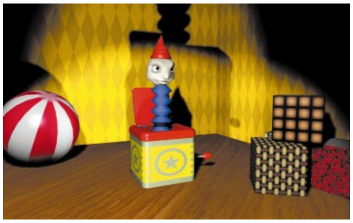
1. A Jack-in-the-box sits quietly in the corner of the little boy's room.



2. The crank starts to rotate and the box begins to bulge in anticipation of something big.



3. Suddenly the box burst open and Jack emerges, bouncing up and down.



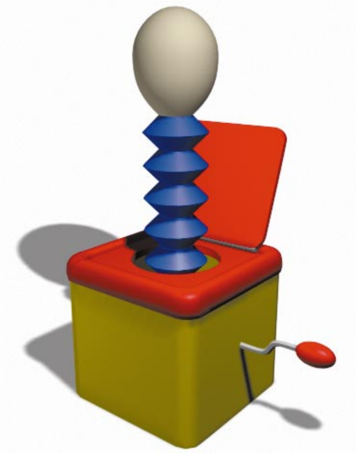
4. Jack settles down to sit quietly in the corner as if nothing had ever happened.

PROJECT TWO

STORYBOARD

7 Curves and Surfaces

This lesson will introduce you to modeling with NURBS (Non-uniform rational B-spline) surfaces. This geometry type lets you create curves and surfaces to build up your models. In this lesson, you will build the jack-in-the-box using tools such as revolve, loft and extrude. You will also begin learning about how to use existing geometry to help you build new geometry.



The Jack in the box

In this lesson you will learn the following:

- How to build a box from a quarter cylinder
- How to attach and close surfaces
- How to work with layers
- How to duplicate curves from an existing surface
- How to draw curves
- How to use surface tools such as loft and revolve
- How to group your objects to prepare for animation

Set up your project

Since this is a new project, you must set a new directory as your project directory. This will let you separate the files you generate in this lesson from other lessons.

1 Set the courseware project

To manage your files, you can set a project directory that contains sub-directories for different types of files that relate to your project.

- Go to the **File** menu and select **Project → Set...**

  A window opens that points you to the Maya projects directory.
- Open on the folder named *learningMaya*.
- Click on the folder named *projectTwo* to select it.
- Click on the **OK** button.

  This sets the learningMaya directory as your current project.

2 Make a new scene

- Select **File → New Scene.**

  This makes sure that your current scene is part of the new project.

BUILDING THE BOX

The box will be built out of several surfaces. You will start by building the base using a primitive cylinder. You will then extract curves from that surface to build the top of the box.

Setting up the Base

You will build the base by first drawing a quarter of a primitive cylinder that will act as one of the corners of the base. This corner will be duplicated to set up all four corners.

1 Create a quarter cylinder

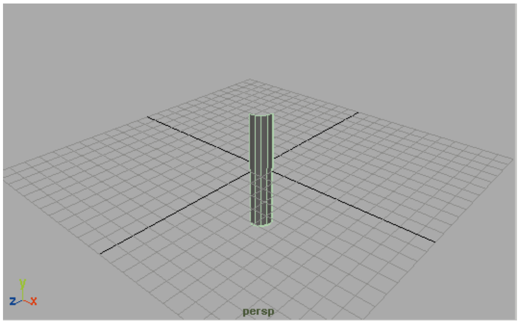
- Select **Create → NURBS Primitives → Cylinder - ❐**.
- In the option window, press the **Reset** button then set the following:

  End Sweep Angle to **90**;

  Ratio of Height... to **8**;

  Number of Sections to **4**.

  This creates a quarter cylinder. This primitive provides you with a starting surface for the box.
- Click **Create** then **Close**.
- Press **3** to increase the surface's smoothness then press **5** to turn on hardware shading.

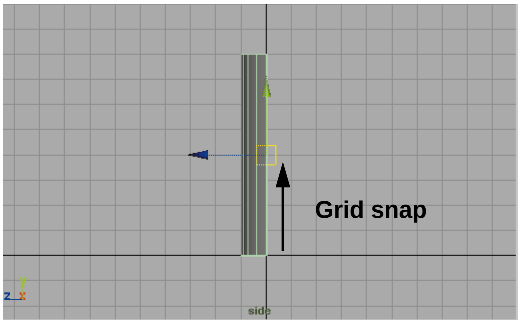


Quarter cylinder

Note: Throughout this lesson, you will want to press 3 to increase the smoothness of any new object. All of the images in this book will reflect this setting.

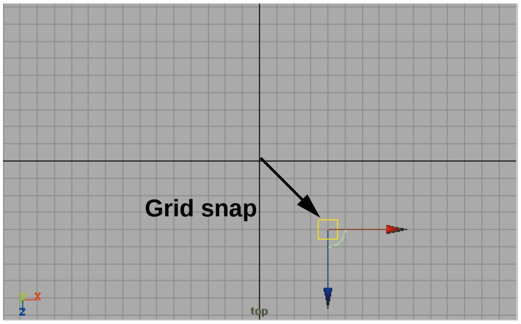
2 Move the cylinder

- Select the **Move** tool.
- From a side view, press the **x** key to use grid snapping then **Move** the cylinder up by 4 units so that it is sitting on the ground.



Side view

- From a top view, use grid snap to **Move** the cylinder diagonally by **4** units in both X and Z.



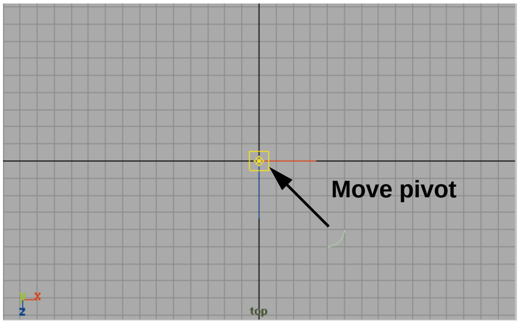
Top view

3 Set the cylinder's pivot

- With the cylinder selected and the **Move** tool active, press the **Insert** key.

This makes the pivot manipulator available. This allows you to edit the object's pivot location.

- In the top view, grid snap the pivot back to the origin.



Relocated pivot

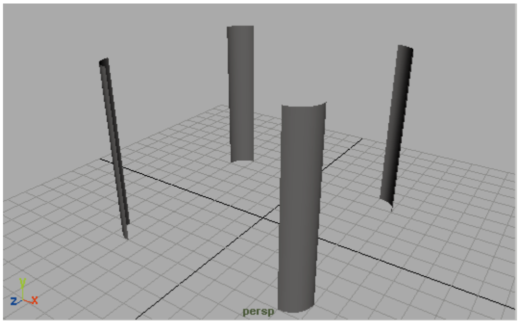
- Again press the **Insert** key to go back to the **Move** tool.

4 Duplicate the cylinder

- Select **Edit** → **Duplicate** - ❑.
- Set the following:

  Rotate Y to **90**;

  Number of Copies to **3**.
- Press **Duplicate**.



Four corners

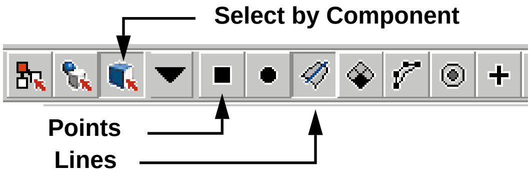
Attaching the corner surfaces

The base will work best if it is a single surface. You will now attach the various corner surfaces to create one complete surface. You will start by selecting isoparms from the existing surfaces. Isoparms are lines that sit on the surface that help define its topology. In some ways they are the grid lines of a NURBS surface.

1 Select an end isoparm

To attach the corners, you need to work with the surface's isoparms. This will require that you go into component selection mode. This will let you select surface isoparms that will help you control the attach.

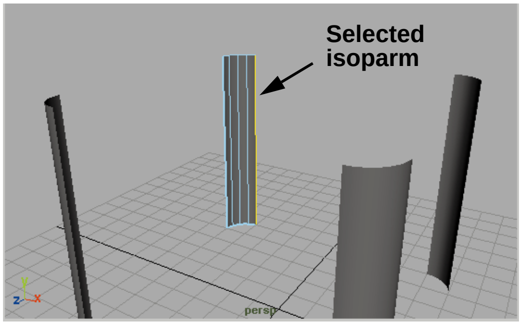
- Press the **Select by Component type** button on the status line.
- Click on the **Points** button to turn it off.
- Click on the **Lines** button to turn it on.



Selection icons

Now you can easily select the edges of quarter cylinders.

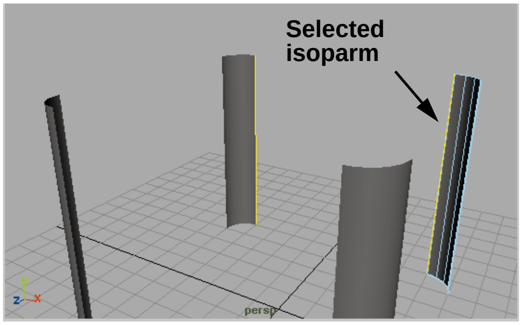
- Click once on one of the corners. This selects the surface and highlights it.
- Click again on the end isoparm. It will highlight in yellow.



Selected isoparm

2 Select the opposing isoparm

- Click on a second corner surface.
- Press the **Shift** key and click on its end isoparm, the one that sits opposite the first isoparm you selected, to add it to the selection.



Second isoparm selected

3 Attach the surfaces

- Go to the **Modeling** menu set.
- Select **Edit Surfaces** → **Attach Surfaces** - ❐.

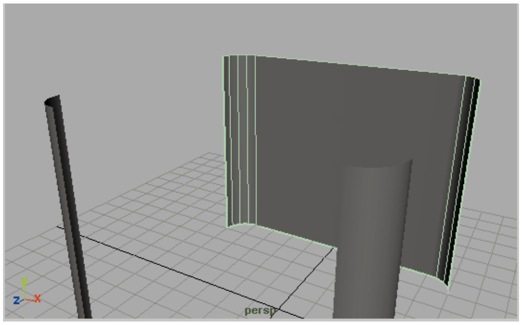
- In the option window, set the following:

  Attach Method to **Connect**;

  Multi Knots to **Keep**;

  Keep Originals to **Off**.

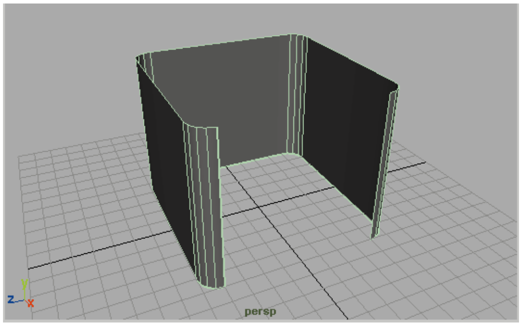
- Click **Attach**.
- Close the tool options window.



Attached surface

4 Attach the other surfaces

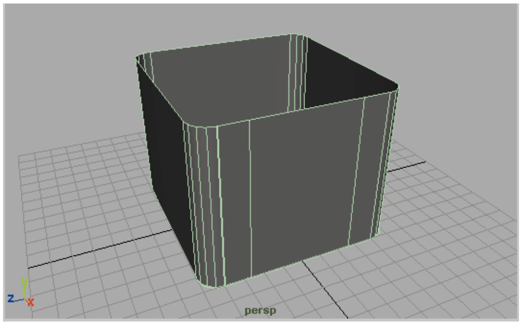
- Select the two opposing isoparms between the newly attached surface and the next corner.
- Select **Edit Surfaces** → **Attach Surfaces**.
- Repeat for the next corner.
- Now you have a single surface but it is not completely closed. Since you cannot attach a surface to itself, you will instead use the close tool.



Single attached surface

5 Close the surface

- Click on the **Select by Object type** button on the Status line.
- Select **Edit Surfaces** → **Open/ Close Surfaces**.
- In the Channel box, rename this surface to *base*.



Closed surface

6 Delete History

- Select **Edit** → **Delete by Type** → **History**.

  This gets rid of any extra nodes so that the model is as light as possible for animation down the line.

PROJECT TWO

7 Save your work

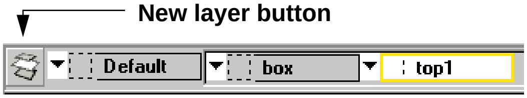
- Select **File** → **Save As** and name the new file *jack_01*.

Creating the top surface

To build the top surface, you will extract an isoparm from the base then use that as one of the generation curves for the top surface.

1 Create two layers

- Select **Options** → **Layer Bar**.
- Click on the New layer button.
- Double-click on the layer then enter the name *box_layer*.
- Click on the New layer button.
- Double click on the layer then enter the name *top_layer*.


New layer button

New layers

2 Assign surface to box layer

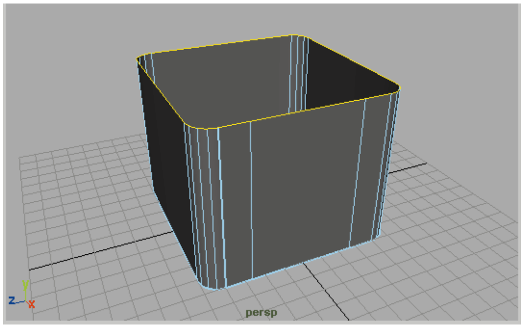
- **Select** the *base* surface.
- Click on the arrow icon on the *box* layer and select **Assign Selected** from the pop-up.

3 Select an isoparm from the base

- Click on the *base* surface with the right mouse button and select **Isoparm** from the marking menu.

  This menu lets you go to the surfaces component level without changing selection modes.

- Click on the top edge of the surface to select this isoparm.

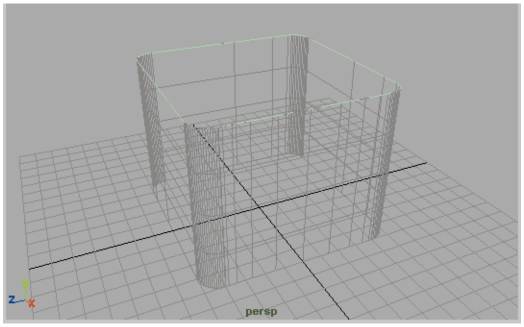


Selected curve

4 Duplicate the isoparm

- Select **Edit Curves** → **Duplicate Surface Curves**.
- Click on the arrow icon on the *top_layer* and select **Assign Selected** from the pop-up.
- Click on the arrow icon on the *box_layer* and select **Template**.

  Now you can focus on the top of the box while keeping the base around as a visual reference.

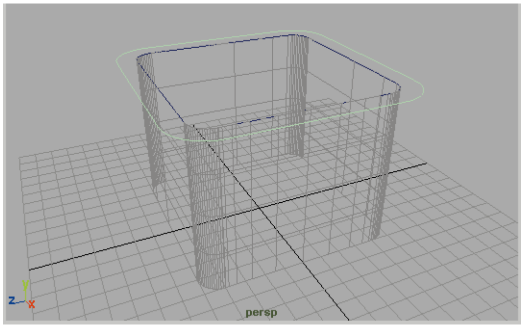


Duplicated curve

5 Offset the curve

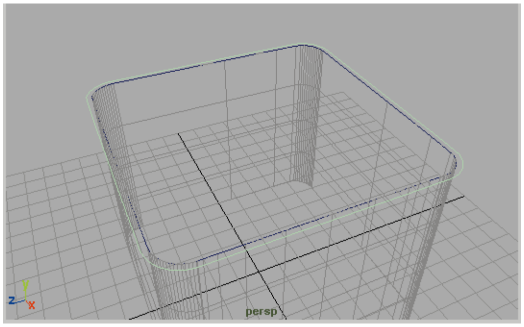
- With the duplicated curve still picked, select **Edit Curves** → **Offset** → **Offset Curve**.

This creates a second curve that is offset from the original. The default offset seems a little big.



Offset curve

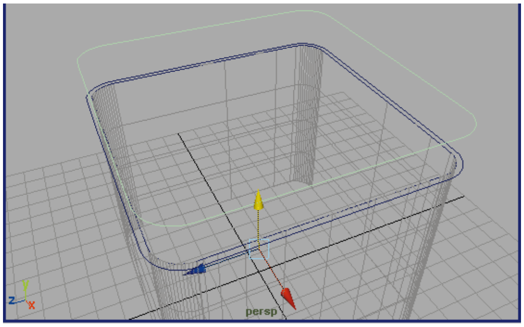
- Click on the *offsetCurve* input node in the Channel box.
- Set the **Distance** to **0.2**.



Offset curve

6 Duplicate the curve

- Select **Edit → Duplicate - ❐**.
- In the Option window, click on the **Reset** button then click **Duplicate**.
- **Move** the new curve up along the Y axis to create the depth of the box's top surface.



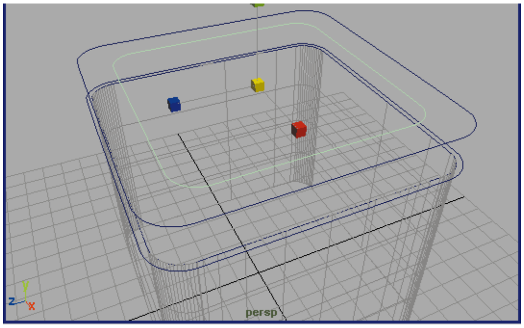
Duplicated curve

7 Create another duplicate curve

- Select **Edit → Duplicate**

  You can see that the pivot of the curve lies at the origin. It would be more useful if the pivot was at the center of the curve.

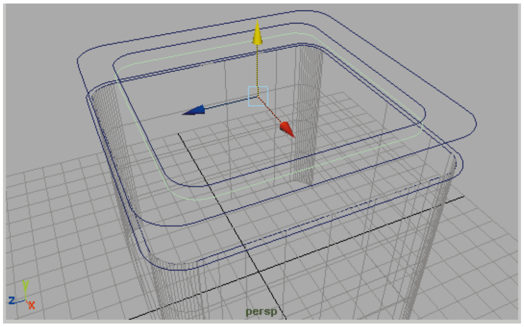
- With the curve selected, select **Modify → Center Pivot.**
- **Scale** the new curve in a little to create a border.



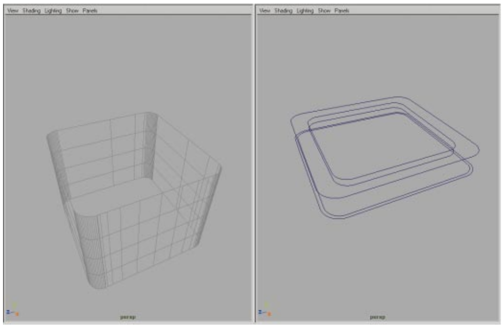
Scaled curve

8 Create another duplicate curve

- Again, select **Edit → Duplicate**
- **Move** the new curve down.



Duplicated curve



Side-by-side layout

Creating the top edge surface

You will now build the top of the box using these curves to create several lofted surfaces that will then be attached with blending. In the end you will have two surfaces that will make up the top of the box.

1 Set up your view panels

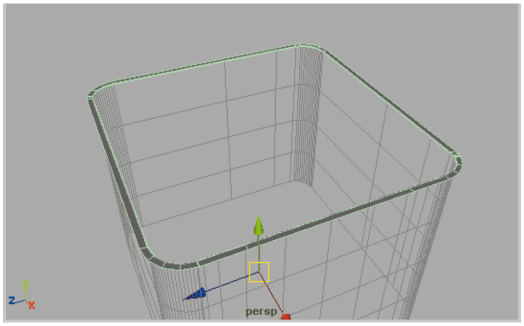
- Select **Panels → Layouts → 2 Side by Side**.
- Set up both panels as perspective views.
- For the first panel, select **Show → None** then **Show → Curves**.
- For the second panel, select **Show → None** then **Show → Surfaces**.
- Select **Show → Grid** to turn off the grid for both the view panels.

  You will use one of the panels for selecting curves and the other to preview the resulting surfaces. This is a useful technique for easily picking different object types.

2 Create a lofted surface

- In the perspective panel that only shows curves, **select** the first and second curves that you created.
- Select **Surfaces → Loft**.

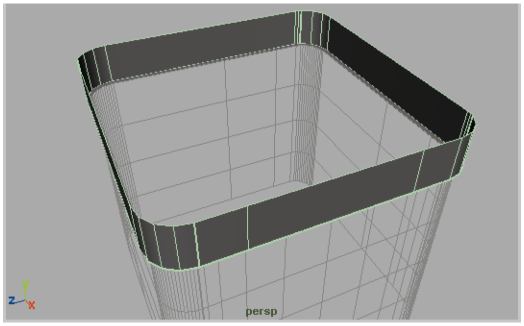
  A surface is generated that uses the curves as edge isoparms.



Lofted surface

3 Create a second lofted surface

- **Select** the second and third curves.
- Select **Surfaces → Loft**.



Lofted surface

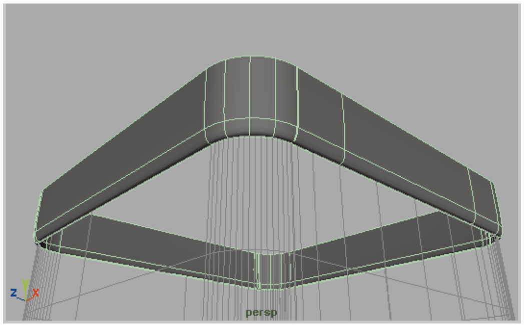
4 Attach the surfaces using blend

- **Select** the two surfaces in the perspective panel that only shows surfaces.
- Select **Edit Surfaces → Attach Surfaces - ❐**.
- In the option window, set the following:

  Attach Method to **Blend;**

  Keep Originals to **Off**.
- Click **Attach**.

  Now the two surfaces are blended together smoothly.

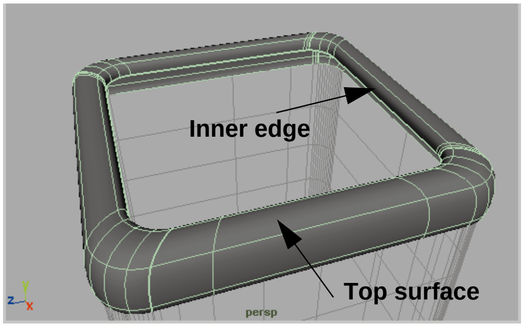


Attached surfaces

5 Complete the shape

- Create a third lofted surface using the methods taught above. This surface will be the top surface.
- **Attach** the surfaces.
- Repeat to create another lofted surface for the last two curves. This surface will be the inner edge.
- **Attach** these together using the method taught with the corner surfaces.

  You will need to select the edge isoparms using the right mouse button to make sure that you attach the surfaces properly.



Final shape

6 Delete History

- **Select** the surface.
- Select **Edit → Delete by Type → History**.

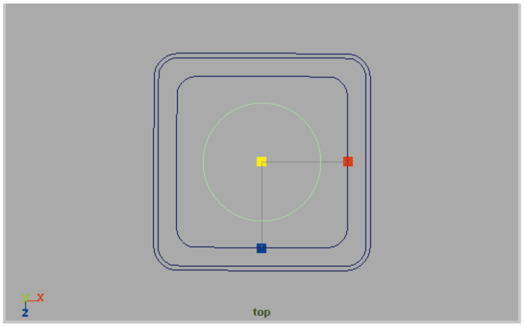
  Again, you don't want the history to interfere with your animation later.

Creating middle section

To create the middle section of the top, you will loft the last of the existing curves with a primitive circle. The circle will act as the hole from which Jack will emerge.

1 Create and place a circle primitive

- Select **Create → NURBS Primitives → Circle**.
- In a top view, **Scale** the circle to fit in the middle of the top surface.

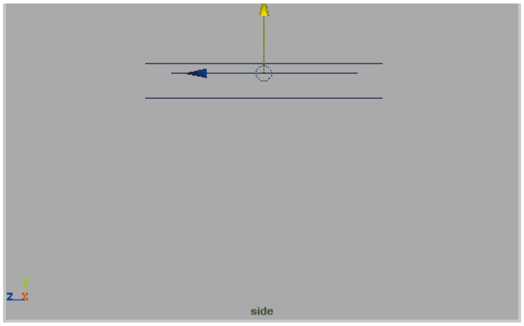


Circle

2 Snap-Constrain the circle

- In a side view, click on the Y-axis handle to activate it.
- Select the **Move** tool.
- Place your cursor over the curve that you want to match the circle to.
- Press the c key to use curve snapping then *with your middle mouse button* click-drag on the curve.

  This snaps the circle so that it only moves along the Y axis while it snaps to the curve.

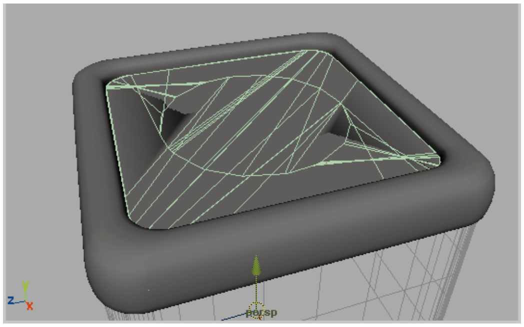


Curve snapping the circle

3 Loft between the curves

- Return to your two perspective layout.
- **Select** the last curve and the circle.
- Select **Surfaces → Loft**.

  The resulting surface is twisted. The circle is not oriented correctly.



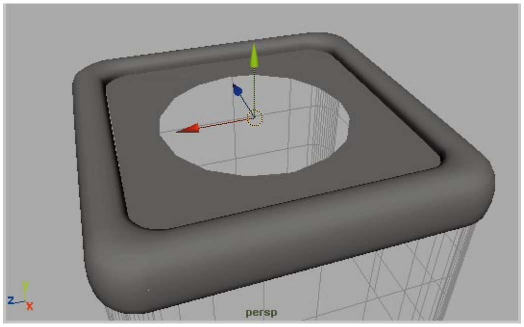
Strange lofted surface

4 Update the surface using history

You can update the surface by editing the profile curves. If you rotate the circle then you can remove the twisting.

- **Select** the circle.

- In the Channel box, set **Rotate Y** to either **90** or **-90,** whichever corrects the surface.

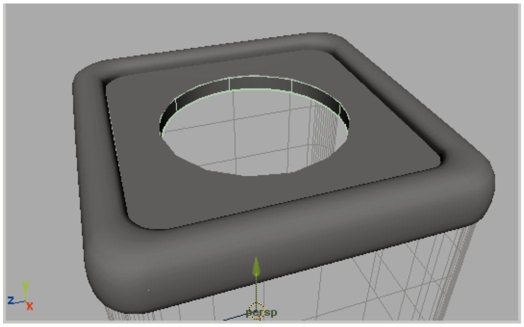


Corrected loft

5 Duplicate and position the circle

- **Select** the circle.
- **Duplicate** the circle.
- **Move** the circle down just below the first.
- Select the two circles.
- Select **Surfaces** → **Loft**.

  You will keep these surfaces as two separate surfaces instead of attaching them together.



Inner ring

6 Delete History

- **Select** the two new surfaces.
- Select **Edit** → **Delete by Type** → **History**.

7 Add the surfaces to the box layer

- **Select** the three top surfaces.
- From the *box_layer's* pop-up menu, select **Assign Selected**.

  They will be templated to match the chosen layer state.

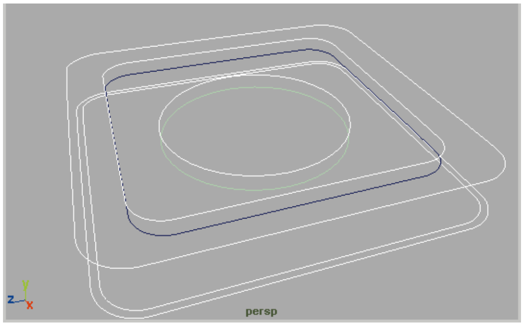
Creating the lid

The box's lid will be constructed using one of the existing curves that will be turned into a Set planar surface.

This surface will then be duplicated and the two surfaces will be joined using another lofted surface.

1 Set up the starting curve

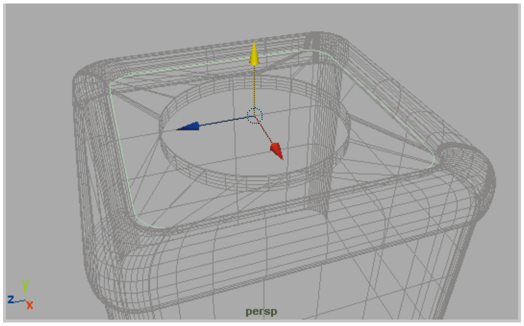
- **Select** all the curves except the last rounded curve that was created before the circle.



Selected curves

- Press the **Backspace** key to delete them.

- In the panel that only displays curves, select **Show → Surfaces**.
- Select the remaining curve.
- **Scale** it a little bit smaller then **Move** it up slightly above it original position.



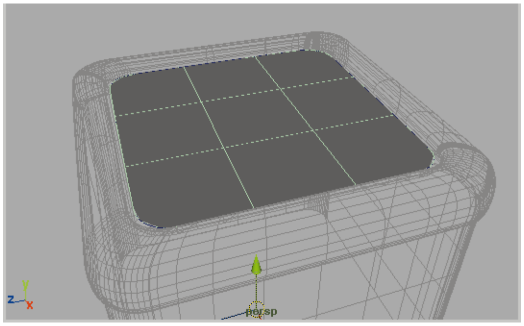
New curve

2 Create a set planar surface

- With the curve selected, select **Surfaces → Planar**.

  This creates a trimmed surface. This means that the surface actually extends outside of the boundary curve then the curve is used to trim the surface.

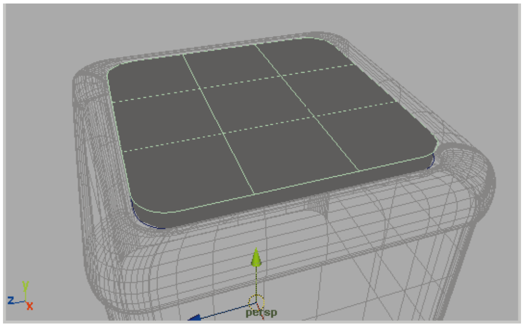
- **Delete History** on the surface



Planar surface

3 Duplicate the surface

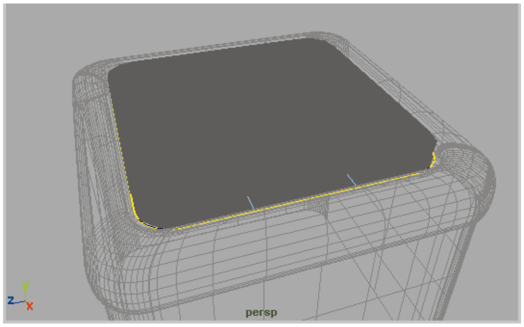
- Select the planar surface.
- Press **Ctrl-d** to create a second surface.
- **Move** the surface above the first surface.



Two surfaces

4 Select the trim edges

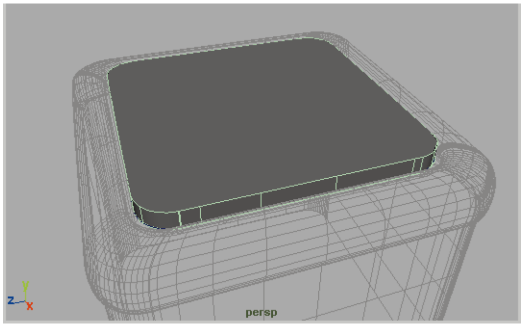
- Click with your right mouse button on the first planar surface and select **Trim edge** from the marking menu.
- Click on the edge of the surface to select the trim edge.



Trim edge

- Click with your right mouse button on the second planar surface and select **Trim edge** from the marking menu.
- Press the **Shift** key and click on the edge of the surface to select the trim edge.
- Select **Surfaces** → **Loft**.

  Very Important! Do not delete history for this surface. The history will play a role later if you decide to deform the planar surfaces.



Lofted surface

The Hinge

To act as the hinge for the box's lid, you will create and position a cylinder. This surface will also act as the parent for the other surfaces to help define the rotation of the lid.

1 Create and place a cylinder

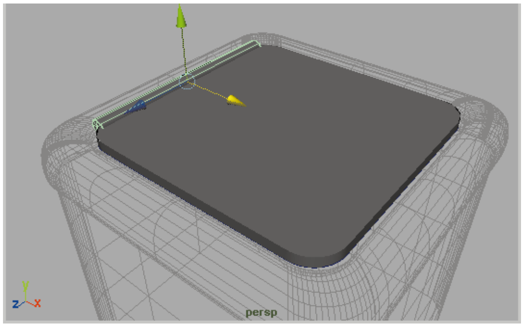
- Select **Create** → **NURBS Primitives** → **Cylinder - ❐**.
- Click the **Reset** button then set the following:

  Axis to **Z**;

  Radius to **0.25**;

  Ratio of Height to **28**;

  Caps to **Both**.
- Click **Create**.
- **Move** the cylinder up to the back of the lid.
- Rename the cylinder to *hinge*.



Hinge in place

2 Freeze the hinge's transforms

- Select the *hinge*.
- Select **Modify** → **Freeze transformations**.

  This sets the transform values to 0, 0, 0 for translation and rotation, and 1, 1, 1 for scaling while leaving the object in its current location. This gives a good starting point for the animation of the hinge.

3 Parent the lid surfaces

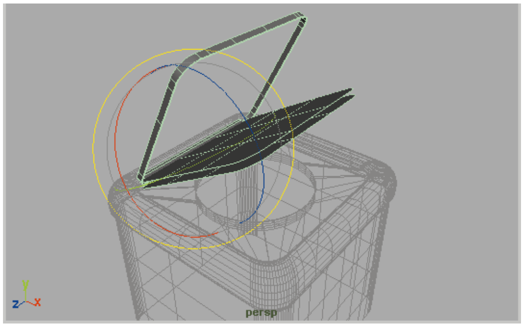
- **Select** the three lid surfaces then press the **Shift** key and click on the *hinge*.

- Press the **p** key to parent them.

4 Test the rotation of the hinge

- **Select** the *hinge*. Notice how the rest of the lid is selected at the same time since they are children of the hinge.
- Select the **Rotate** tool.
- Click drag on the axis manipulator to rotate the lid.

  Oops, the lofted surface seems to be rotating twice as fast as the planar surfaces.

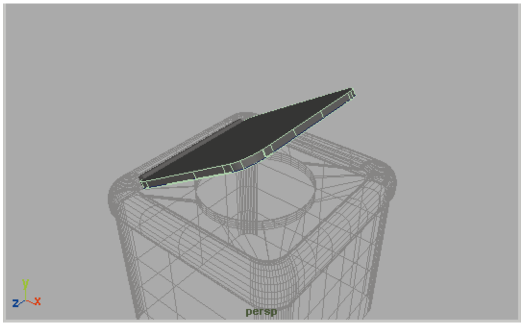


Lofted surface

5 Correct the loft surface

- **Select** the loft surface.
- In the Attribute editor, set **Inherits Transform** to **Off**.

  This tells the surface to not rotate with the hinge. Now it will only rotate as a result of its construction history and you will avoid the double transformations.



Lofted surface

6 Add the new group to box layer

- Select the *hinge*.
- **Assign** the new surfaces to the *box* layer. It will be templated.

The Handle

To create a handle for the jack-in-the-box, you will extrude a circle along a path.

1 Create and place a primitive circle

- Select **Create → NURBS Primitives → Circle - ❐**.
- Click the Reset button then set the following:

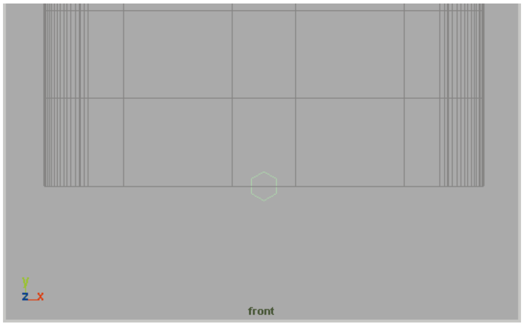
  Axis to **Z**;

  Radius to **0.20**;

  Degree to **Linear**.

  Number of Sections to **6**;

  This Linear setting creates a curve that has flat spans. This will give the look of a faceted metal.

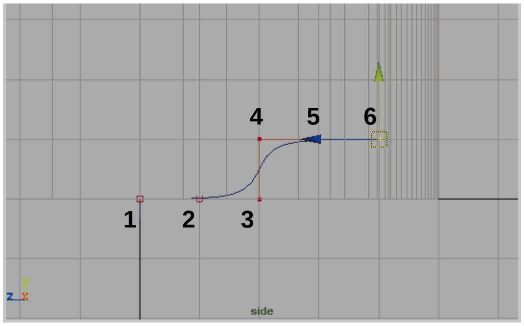


Path curve

2 Draw a path curve

- In a side view, zoom in to the origin.
- Select **Create → CV Curve Tool**.

  Press the **x** key to use grid snapping and draw the following curve using 6 CVs:



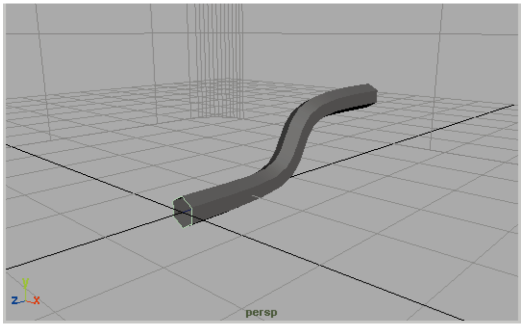
Path curve

- Press the **Enter** key to complete the curve.

3 Extrude the circle

- **Select** the circle and the path curve. Make sure that you select the path curve last.
- Select **Surfaces → Extrude**.

The resulting surface is a result of the circle being passed along the path curve. The resulting surface is faceted (Linear) in one direction and smooth (Cubic) along the other.

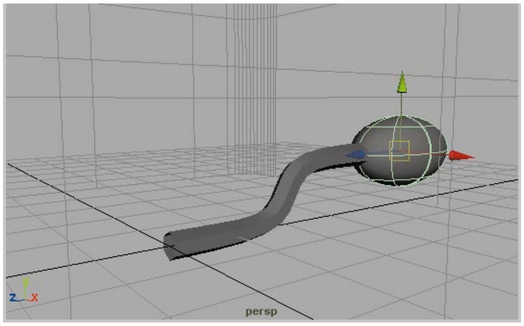


Extruded surface

- **Delete History** from the surface.
- **Delete** the circle and the path curve.
- Rename the surface to *crank*.

4 Add a sphere to the end

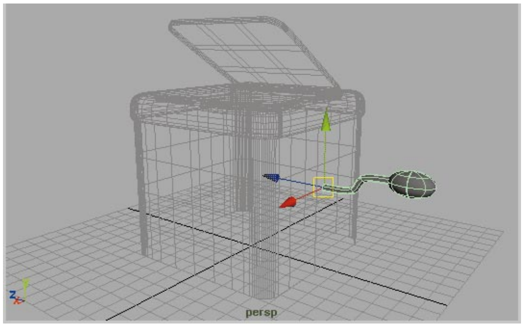
- Create a primitive sphere.
- **Scale** it into an egg-like shape.
- **Move** it to the end of the *crank* handle.
- Press the **Shift** key and add the *crank* to the selection.
- Press the **p** key to parent the *sphere* to the *crank*.



Crank handle

5 Position the crank

- **Move** the crank to the side of the box near the middle. You may need to look at both a top and a side view at the same time to position the crank successfully.



Completed crank

6 Add the crank to box layer

- Select the *crank.*
- **Assign** the new surfaces to the *box_layer*. It will be templated.

Jack's body

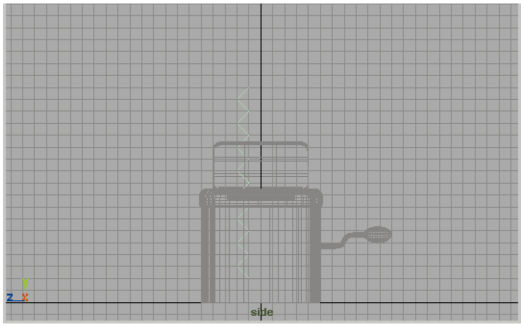
You are going to create an accordion-like surface for Jack's body. This will include a Linear (degree 1) profile curve that is revolved into shape.

1 Create a Linear NURBS curve

Like the profile of the crank, you will create the body surface using a faceted (Linear) curve in one direction and a smooth (Cubic) curve along the other

- Select **Create → CV Curve Tool - ❐.**
- In the Option window, set **Curve Degree** to **1 Linear.**

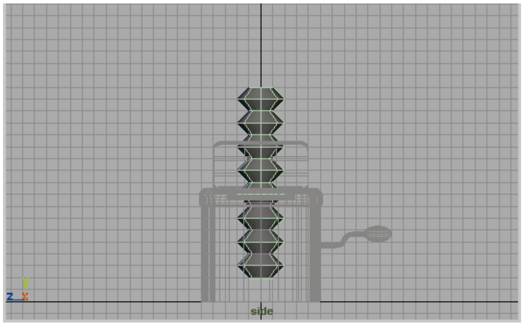
  By using a *linear* setting, a curve will be created with straight lines drawn between CVs.
- In the side view panel, use grid snapping to help you draw a zig-zag curve to act as the profile for the body surface.



Profile curve

2 Revolve the curve into a surface

- Select **Surfaces → Revolve**.



Revolved body surface

3 Delete history

- **Select** and **Delete** the profile curve.

  By removing the profile curve, you have deleted the surface's history.

Jack's head

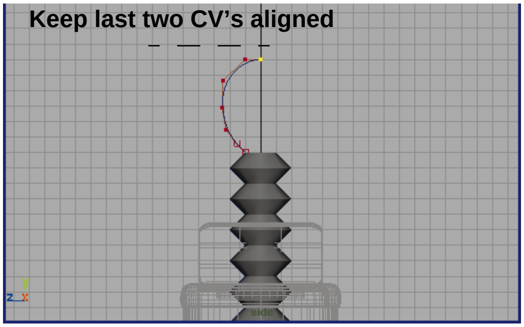
Using a Cubic (degree 3) curve, you are going to create another revolved surface. This surface will start out very simple then it will be used in the next lesson to sculpt facial features for Jack.

1 Create a degree 3 curve

- Select **Create → CV Curve Tool - ❐.**
- In the Option window, set **Curve Degree** to **3 Cubic.**

  The *cubic* setting will create a smooth curve that is interpolated from the CVs.
- Draw a curve from the end of the body surface up to the Y axis line.

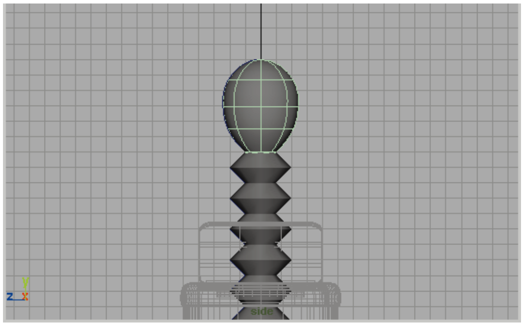
  Use grid snap to make sure that your last two CVs are lined up perpendicular to the Y axis line. This will prevent Jack's head from being pointy when it is revolved.



CV curve

2 Revolve the curve into a surface

- Select **Surfaces → Revolve**.



Revolved head surface

3 Delete history

- **Select** and **Delete** the profile curve.

Adding some color

In another lesson, you will set up the rendering for the Jack-in-the-box. For now, you will create a few shaders and assign them to the parts of the model to

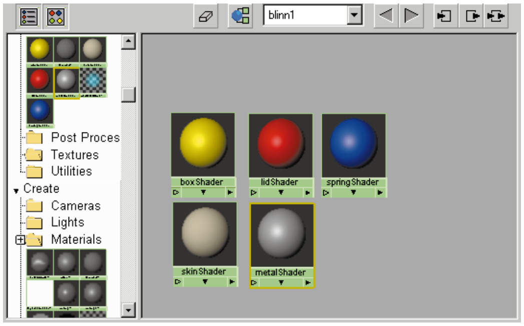
1 Reveal the box layer.

- From the box layer's pop-up, select **Standard**.

  Now the objects are visible in the shaded view.

2 Create and assign color shaders

- Use the methods you learned in Project one to create five Blinn materials for the scene.
- Edit the color of the shaders as shown below:



New shaders

- Name the shaders as shown below and assign them to the following surfaces:

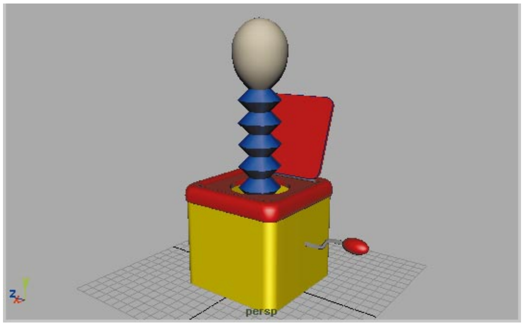
  baseShader to base surface;

  lidShader to lid, box top and the crank handle;

  springShader to Jack's body;

  skinShader to Jack's head;

  metalShader to the hinge and the crank stem.



Box with shaders

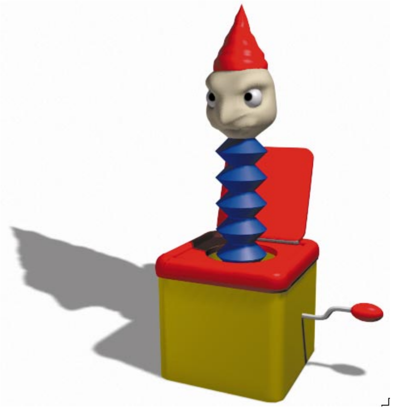
Conclusion

You now have some experience working with curves and surfaces. You have seen how profile curves can be used to generate surfaces with the help of tools such as revolve, loft and extrude.

In the next lesson, you will sculpt Jack's face by manipulating the surface at a component level. You will also learn about Maya's Artisan functionality that lets you mold the surface using a brush-like interface.

8 Organic Modeling

In this lesson, you will sculpt facial features onto Jack by manipulating the surface's control vertices (CVs). At first you will use the transform manipulators to edit the CVs, then you will put Maya's Artisan tools to work using a brush paradigm as you push, pull, and smooth the surface.



Jack's sculpted face

Maya's Artisan tools offer a different approach to how you work in Maya. Rather than typing in values, Artisan lets you use a paint-like interface to edit shapes in Maya. In later lessons, you will also learn how to use Artisan for other important functions.

In this lesson you will learn the following:

- How to rebuild a surface
- How to pull CVs using the Move tool's Normal setting
- How to sculpt surfaces by painting with Artisan
- How to use different brush operations

Getting started

For this lesson, you will continue to work with the model you produced in the last lesson. You are going to work on the face area to define Jack's character.

Editing Control Vertices

Control Vertices represent points on a surface or a curve that help define its shape. By repositioning these important points, you can sculpt the surface.

1 Rotate the surface

Since the curve you used to revolve the head surface was created on the side, this means that the surface's seam is at that point. You want the seam at the back of the head.

- Select the *head* surface.
- In the Channel box, set **Rotate Y** to **-90**.

  In the shaded view, this didn't seem to change anything. But by moving the seam to the back of the head, you can take advantage of some of Artisan's symmetry tools used later in the lesson.

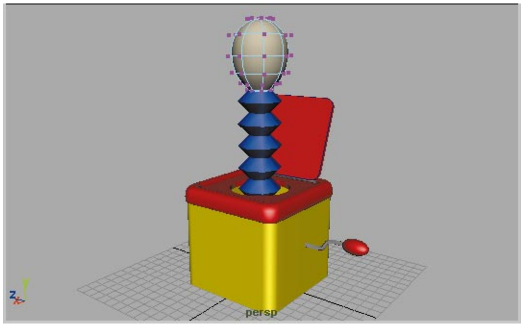
2 Change to component mode

CVs are part of the surface's shape node and can only be selected by going to component selection mode.

- Make sure that **Show → NURBS CVs** is **On** for the perspective panel.
- On the status line, click on the **Select by Component Type** button.
- Turn the **Points** button to **on** and the **Lines** button to **off**.

  Now the CVs are shown on the head's surface.
- **Select** the *base* surface of the box.

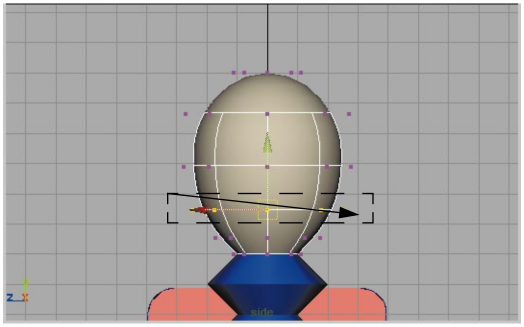
  Now this surface is displaying its CVs. As you select surfaces, you can focus on editing their CVs.
- Again **Select** the *head* surface.



Head surface with CVs visible

3 Select CVs

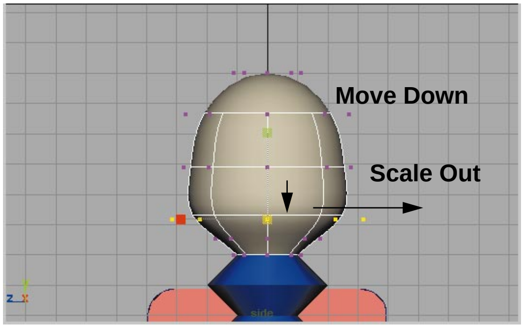
- From a side view, **select** the row of CVs that lie just under the center of the shape.



Edited CVs

4 Move and Scale some CVs

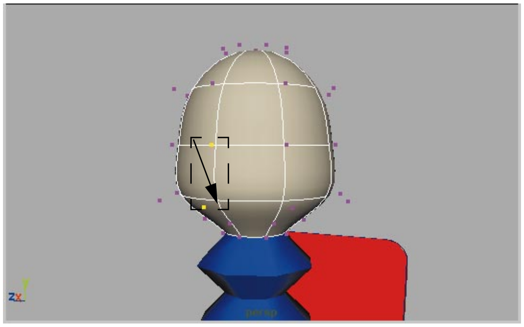
- **Move** the CVs down a little.
- **Scale** the CVs out to tighten the shape.



Edited CVs

5 Focus on a few CVs

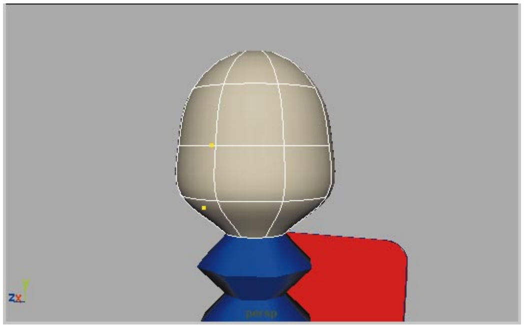
- In the perspective view, **select** the two CVs that lie on and just below the middle of the head on its centerline.
- Be careful that you don't select CVs at the back of the character's head.



Selected CVs

- Select **Display → Hide → Unselected CVs**.

  Now you can only see the selected CVs which will make it easier to work with them.

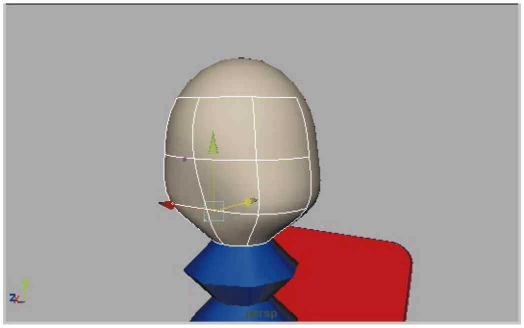


Focusing on the two CVs

6 Move using world space

- **Select** the bottom CV.
- Select the **Move** tool.
- **Move** the CV into the head.

  You are moving the CV in world space using the XYZ coordinates to guide you.



Moving the bottom CV

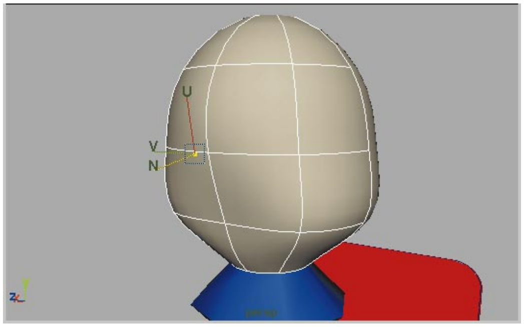
7 Move CV normal

Sometimes moving a CV in world space can be difficult to control. You will now use the **Move** tool's **Normal** setting to work more closely with the surface.

- **Select** the upper CV.
- Double-click on the **Move** tool to open the Tool options.

- Set your **Move Options** to **Normal**.

  The manipulator now displays new axes that are labelled U, V and N. The U and V labels represent the two dimensions of the surface as represented by the isoparm lines. The N represents the normal direction off of the surface.

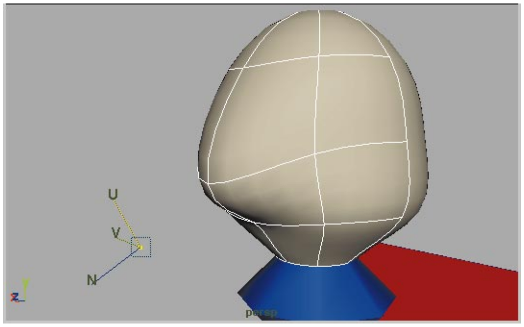


Move CV normal manipulator

- Drag on the **N** axis of the manipulator to **Move** the CV out of the head.

  You are moving the CV in world space using the XYZ coordinates.

- Drag down on the **U** axis of the manipulator to **Move** the CV down along the surface of the head.



Moving the top CV normal

  Now the CV is moving along the shape of the surface.

8 Return to object mode

- Press **F8** to go back to object mode.

  Oops, there is a warning in the feedback line. It is telling you that the **Move → Normal** option does not work in Object mode.

- Double-click on the **Move** tool to open the Tool options.
- Click the **Reset** button to go back to the default settings.

9 Save your work

- Select **File → Save Scene As...**
- Enter the name *jack_02* and **Save**.

USING ARTISAN

You could now continue pulling CVs until you have sculpted a head out of this surface. Instead you will use Maya's Artisan tools. These will let you push and pull on the surface as if it were a piece of clay.

Setting up the surface

To work effectively with Artisan tools it is important that the surface has a large number of CVs. You must therefore rebuild the surface so that it has more available CVs for Artisan to work with. You will also change the material qualities on the head to show stronger highlights. This will help you work in hardware shading when using Artisan.

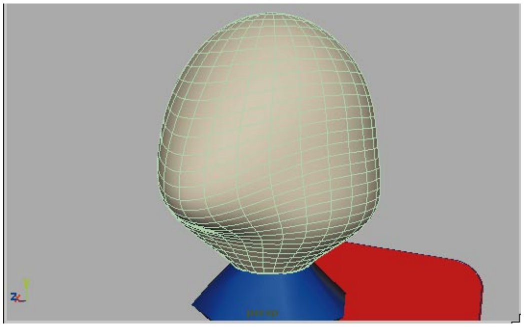
1 Rebuild the surface

- Select the *head* surface.
- Select **Edit Surfaces → Rebuild Surfaces - ❐**.

- In the option window, click the **Reset** button then set the following:

  Number of Spans U to **30**;

  Number of Spans V to **30**.



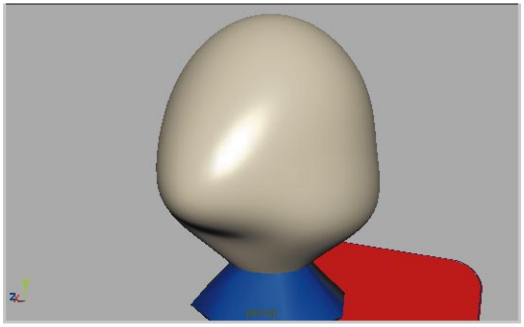
New surface topology

2 Edit the surface's material

- Open up a **Hypershade** panel.
- With the *head* selected, select **Graph → Graph Materials on Selected Objects**.
- Double click on the icon to open the Attribute editor.
- Set the following material attributes:

  Type to **Blinn**;

  Eccentricity to **0**;



Updated material node

This puts a strong highlight on the surface that will help evaluate the Artisan sculpting.

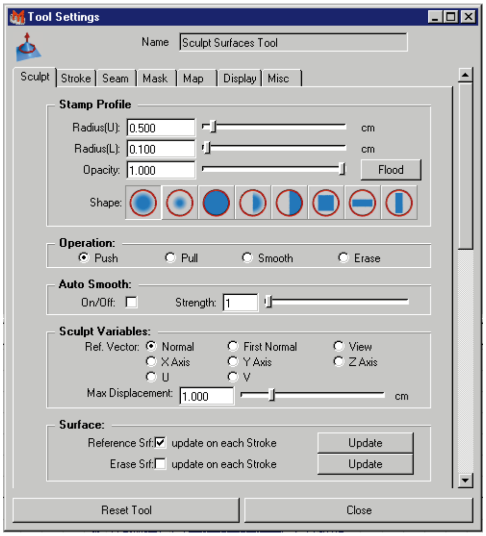
Start sculpting the surface

You will now sculpt the surface of the mask by painting on it using the Artisan Sculpt Surfaces tool. At first, you will just test the tools to get a feel for them. Later, you will erase these brush strokes so that you can paint real facial features.

1 Open the Tool Settings window

- **Select** the *head* surface.
- Select **Edit Surfaces → Sculpt Surfaces Tool - ❐**.

  This opens the Tool Settings window which includes all of Artisan's sculpting options.



Name: Sculpt Surfaces Tool

Stamp Profile

Radius(U): 0.500 cm

Radius(L): 0.100 cm

Opacity: 1.000

Shape:

Operation: ● Push ○ Pull ○ Smooth ○ Erase

Auto Smooth: On/Off: □ Strength: 1

Sculpt Variables:

Ref. Vector: ● Normal ○ First Normal ○ View ○ X Axis ○ Y Axis ○ Z Axis ○ U ○ V

Max Displacement: 1.000 cm

Surface:

Reference Srf: ☑ update on each Stroke

Erase Srf: □ update on each Stroke

Tool Settings window

- Click on the **Reset Tool** button to make sure that you are starting with Artisan's default settings.

- Set the following attributes:

  Under **Stamp Profile**:

  Radius (U) to **0.3**;

  Under **Sculpt Variables:**

  Max Displacement to **0.8**.

- Place the new window to the right of the mask model.

  You will work with this window open, until you are familiar with how to use hotkeys.

2 Paint on the surface

- Move your cursor over the *head* surface. The cursor icon changes to show an arrow surrounded by a red circular outline. The arrow indicates how much the surface will be pushed or pulled while the outline indicates the brush radius.

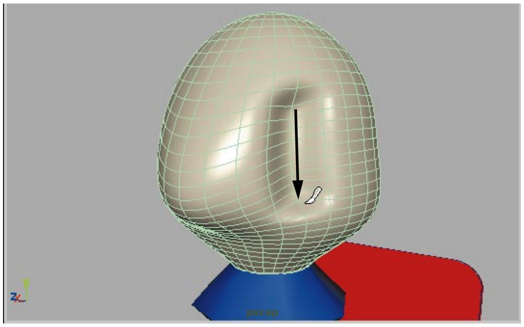
  Artisan's brush icon is context sensitive. It changes as you choose different tool settings.

- Click-drag on the surface. You are now *painting* on the surface to sculpt it.

  You can either click-drag using a mouse, or preferably, draw directly with a stylus.

Tip: Artisan works more intuitively with a tablet and stylus, since the input device mimics the use of an actual paintbrush.

This brush stroke pushes in the surface. The brush stroke is sculpting the surface.



First brush stroke

3 Change the Artisan display

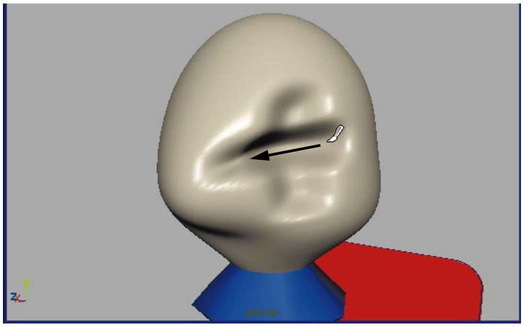
- Click the **Display** tab in the Tool Settings window.
- Click on **Show Active Lines** to turn this option **Off**.

  Now you can focus on the surface without displaying isoparm lines.

4 Paint another stroke

- Paint a second stroke across the mask surface.

  Now it is easier to see the results of your sculpting.



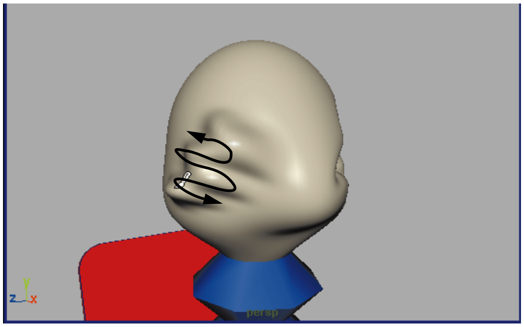
Second brush stroke

The sculpting tools

You will now explore some of the Artisan sculpting operations to see how they work. So far, you have been pushing on the surface. Now you will learn how to pull, smooth and erase.

1 Pull on the surface

- In the Tool Settings window, click on the **Sculpt** tab.
- Under **Operation**, click on **Pull**.
- Tumble around to the other side of the model.
- Paint on the surface to create a few strokes that pull out.

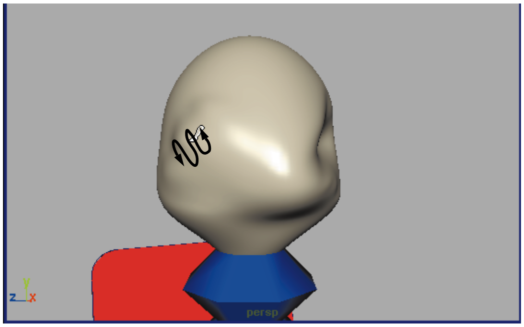


Pulling the surface with several brush strokes

2 Smooth out the results

- Under **Operation**, click on **Smooth**.
- Under **Stamp Profile**, change the **Radius (U)** to **0.6**.

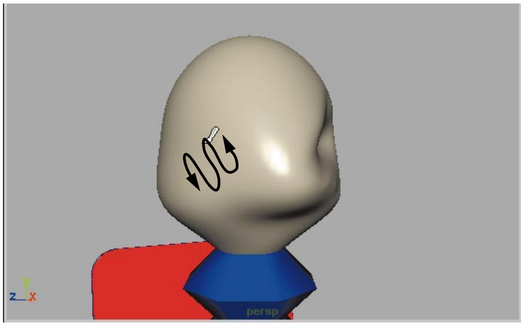
  This increases the size of your brush. You can see that the red outline has increased in size. This is the brush feedback icon.
- Paint all of the strokes to smooth the details. If you stroke over an area more than once, the smoothing becomes more evident.



Smoothing the brush strokes

3 Erase some of the brush strokes

- Under **Operation**, click on the **Erase** option.
- Paint along the surface to begin erasing the existing sculpt edits. Don't erase all the edits.

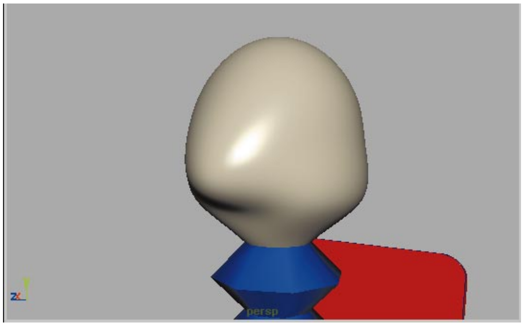


Erasing the brush strokes

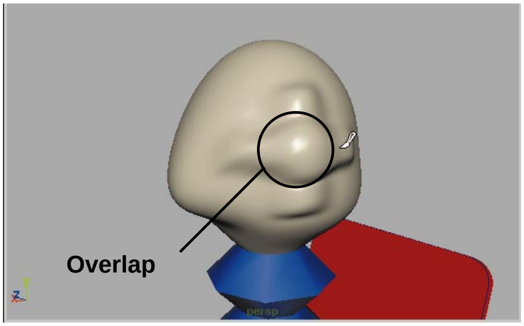
4 Flood erase the surface

- In the **Stamp Profile** section, click on the **Flood** button next to **Opacity**.

  This uses the current operation and applies it to the whole surface using the current opacity setting.



Fully erased surface



Painting with reference update

Updating the reference surface

When you paint in Artisan, you paint in relation to a *reference surface*. By default, the reference surface updates after every stroke so that you can build your strokes on top of each other. You can also keep the reference surface untouched until you decide to update it manually.

1 Change the brush attributes

- Under **Operation**, click on **Pull**.
- Set the following attributes:

  Under **Stamp Profile**:

  Radius (U) to **0.5**;

  Under **Sculpt Variables:**

  Max Displacement to **0.4**.

2 Pull the surface with two strokes

- Paint on the surface to create two crossing strokes that pull out.

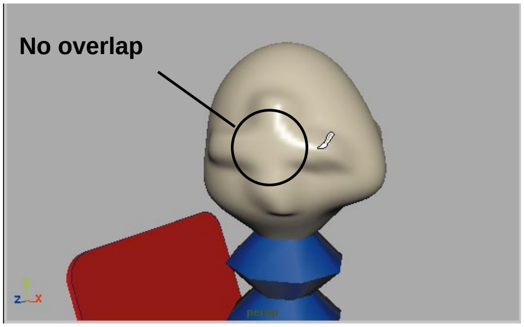
  The second stroke built on top of the first stroke. Therefore, where the two strokes intersect, the height of the pull is higher.

3 Change the reference update

- In the Tool Settings window, scroll down to the **Surface** section, and click on the **Reference Srf: update on each Stroke** to turn this option **Off**.

4 Paint more overlapping strokes

- Paint on the surface to create a few strokes that pull out. This time, the strokes do not overlap. The reference surface does not update, therefore the strokes can only displace to the **Maximum Displacement** value. You cannot displace beyond that value until you update the reference surface.



Painting with no reference update

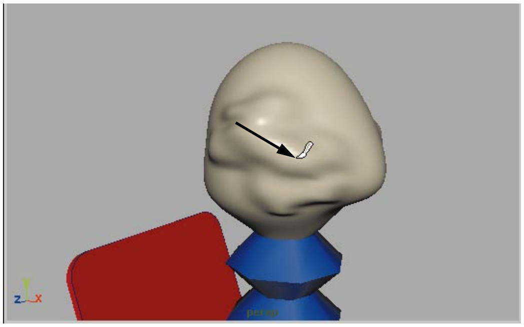
5 Update the reference layer

- In the **Surface** section, click on the **Update** button next to **Reference Srf**.

6 Paint on the surface

- Paint another stroke over the last set of strokes.

  The overlapping strokes are again building on top of each other.



Painting on updated reference layer

7 Flood erase the surface

- Under **Operation**, click on the **Erase** option.
- Click on the **Flood** button.

Sculpting the eye sockets

Now that your surface is clean again, you can sculpt two eye sockets. Ideally, you want the strokes you apply to the left eye to be mirrored on the right eye. Artisan makes painting symmetrical strokes very simple.

1 Turn reference updating back on

- In the **Surface** section, click on the **Reference Srf: update on each Stroke** to turn this option to **On**.

  This will let you build up the facial details as the reference surface updates.

2 Change your stylus pressure setting

If you are working with a tablet and a stylus, you can set up your stylus to create more subtle results.

- Click on the **Stroke** tab.
- Under **Stylus Pressure**, click on **Radius**.

  This means that the harder you press, the larger the radius of the brush stroke. You can now set the upper radius (**Radius (U)**) and lower radius (**Radius (L)**) sizes that will be used by the stylus.

  As you work through this lesson, explore the various stylus settings to see how they can help you paint.

3 Change your Stamp Profile settings

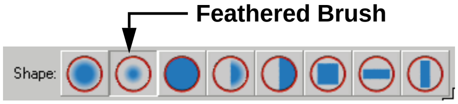
- Click on the **Sculpt** tab.
- Change the **Radius(U)** to **0.7**.
- Change the **Radius(L)** to **0.3**.

  This means that the lowest your stroke will go is 0.3 and the largest will be 0.7.

- Change the **Opacity** to **0.2**.

  This means that each stroke will only have 0.2 of the effect. This lets you use softer strokes to build up a shape.

- Change your stamp setting to the second icon, which has more feathering at the brush's edge.

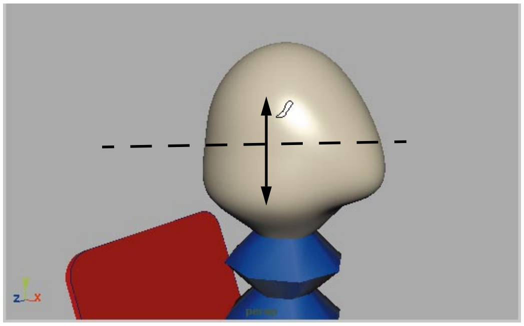


New brush shape

4 Turn Reflection on

- Click on the **Stroke** tab.
- Under **Reflect Paint**, set **Reflection** to **On**.

  By default, it is set to reflect around the U-axis.
- Move your cursor over the surface. You can see that the reflection is going the wrong way.

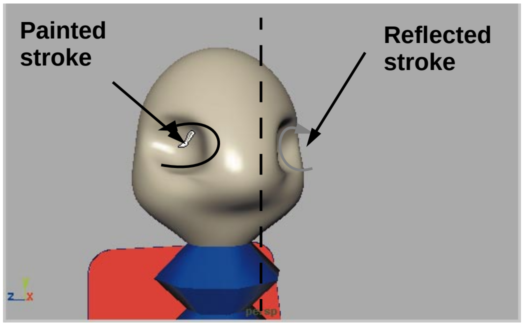


Reflecting in U

- Click on **V Isoparm** for reflecting. Now the two stroke icons are visible on the two sides of the face.

5 Push the eye sockets in

- Press the **u** key and select **Push** from the Artisan marking menu.
- Paint one of the eye socket areas to push it in. The other socket is pushed by the reflection.



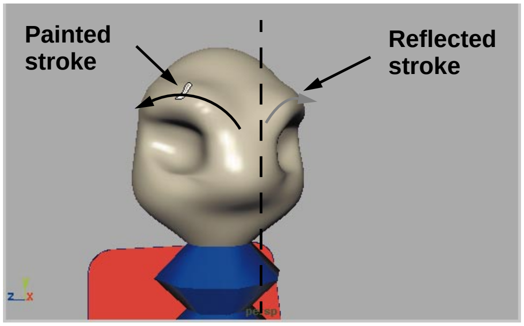
Pushing in eye sockets

6 Pull the eyebrows out

- In the **Sculpt Variables** section, change the **Ref. Vector** to **Normal**.
- Press the **u** key and click to bring up the Artisan marking menu.

  This marking menu is context sensitive and changes depending on which Artisan tool you are using.
- Select **Pull** from the marking menu.

  This is an alternative method for changing the tool operation without using the Tool Settings window.
- Paint the eyebrow areas to pull them out.



Pulling out eyebrows

7 Flood smooth the surface

- Press the **u** key and select **Smooth** from the Artisan marking menu.
- Click on the **Flood** button two times.

8 Save your work

- Select **File** → **Save**

Sculpting the Nose

You will now sculpt the nose. You will work with the Artisan hotkeys to change brush attributes.

1 Change the brush radius

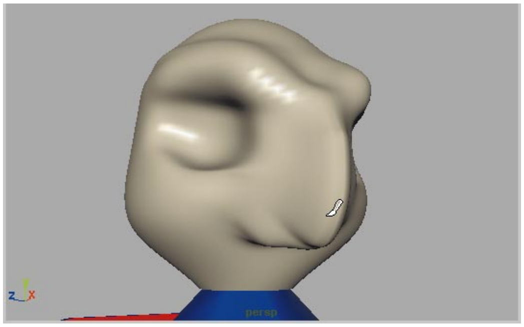
- Press the **b** key and click-drag to make the upper radius smaller. This hotkey gives you quick access to the brush radius.

Note: Artisan has a couple of hotkeys that make it easier to edit key brush attributes. These include:
b - **Radius (U)**
m - **Maximum Displacement**

2 Paint the length of the nose

- Press the **u** key and select **Pull** from the Artisan marking menu.
- Paint from the top of the nose to the tip. Use several strokes to build up the bridge of the nose and the nostril areas. You may need to tumble your view to complete these strokes.

  If you don’t like any of your strokes, you can undo the action using the **z** key, or you can **Erase** back to the original surface. Artisan gives you this flexibility in order to let you explore design alternatives.



Sculpting the nose

3 Flood smooth the shape

Since these strokes appear somewhat bumpy, you can smooth all the strokes using the Flood button.

- Press the **u** key and select **Smooth** from the Artisan marking menu.
- In the Tool Settings window, click on **Flood** twice.

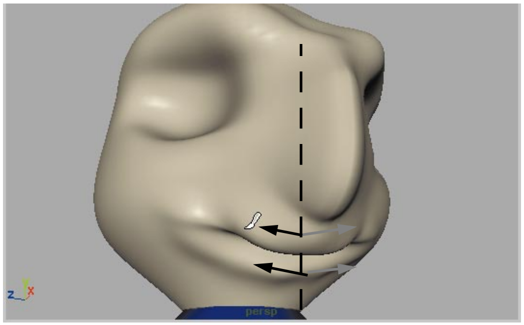
  This smooths out some of the bumps. Because the opacity setting is set to 0.2, the smoothing is more subtle. It is a good idea to smooth your shape regularly to clean up your strokes.

Sculpting the mouth

You will now sculpt the mouth by pulling out the lips.

1 Pull out two lips

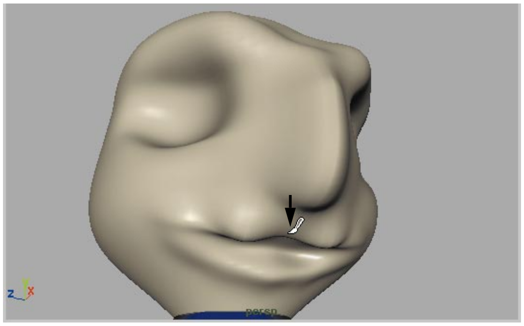
- Press the **u** key and select **Pull** from the Artisan marking menu.
- **Paint** the lip area to pull two lips out.



Pulling out two lips

2 Add the upper lip detail

- Set your **Radius (U)** and **Radius (L)** to **0.1**.
- Press the **u** key and select **Push** from the Artisan marking menu.
- Paint the area above the upper lip to push it in.



Pushing in the upper lip

3 Continue sculpting

- Keep working with the Artisan tools until you get the face that you want.

  Remember to adjust the upper and lower radius of your brush as well as your maximum displacement.

Finishing touches

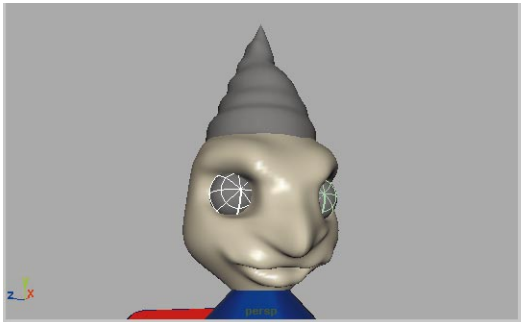
To complete the head, you will add a hat and two eyes. These extra surfaces can be parented into the head.

1 Add a hat

- Now add a primitive **Cone** for a hat and **Move** it to the top of Jack's head.
- Use Artisan's tools to add some detail to the *hat*.
- **Select** the *hat* then select the *head* surface.
- Press **p** to parent the hat to the head.

2 Create eyes for Jack

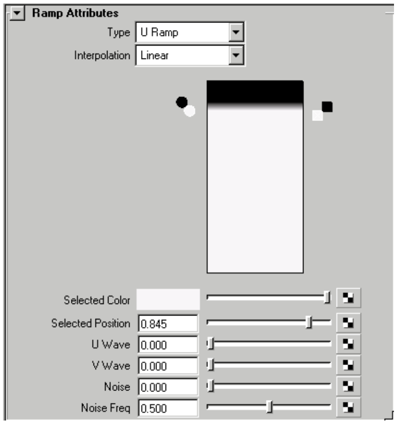
- Add two primitive **Spheres** for eyes and **Move** them into the eye sockets.
- Make sure that the *eye* spheres have a **Rotate X** of -90 so that their polar axis is pointing out from the head.
- **Select** the *eyes* then select the *head* surface.
- Press **p** to parent the eyes to the head.



Head with eyes and hat

3 Add Materials to the eyes

- Open the Hypershade
- Create and apply a *phong* material to the eyes.
- Double click on the new material node.
- Click on the **Map** (checker) button next to **Color**.
- Select **Ramp** from the Create Render Node window.
- Click on the square icon next to the middle color indicator. This deletes this marker.
- Click on the round icon next to the top color indicator.
- Click on the color swatch next to **Selected color** then change the color to black.
- Use the same technique to change the lower color indicator to white.
- Click drag on the round icons to move the two colors to the top of the ramp.



Ramp Attributes

Type U Ramp

Interpolation Linear

Selected Color

Selected Position 0.845

U Wave 0.000

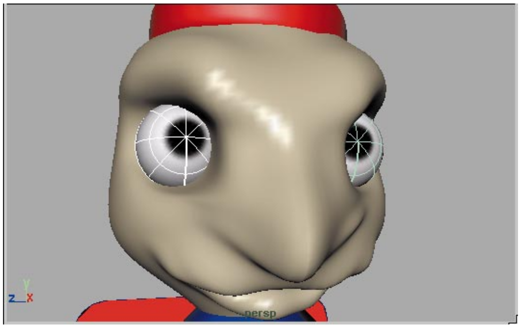
V Wave 0.000

Noise 0.000

Noise Freq 0.500

Ramp

If this doesn't look accurate on your model then the black bar may need to be at the bottom of the ramp.



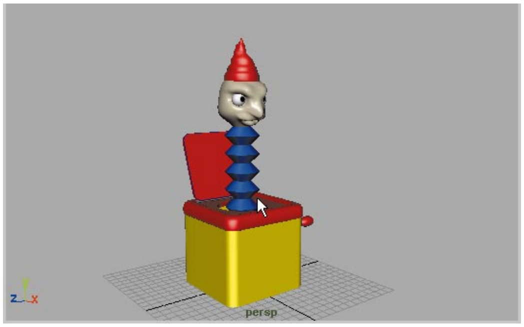
Head with eyes and hat

4 Save your work

- Select **File** → **Save**

Conclusion

You have now edited a surface using its control vertices. You have also used Maya's Artisan tools to help give the face a more organic quality. Generally, you will use CV pulling to get the basic shape of a surface then use Artisan to give it a more natural look. In the next lesson, you will begin to animate Jack and his box.

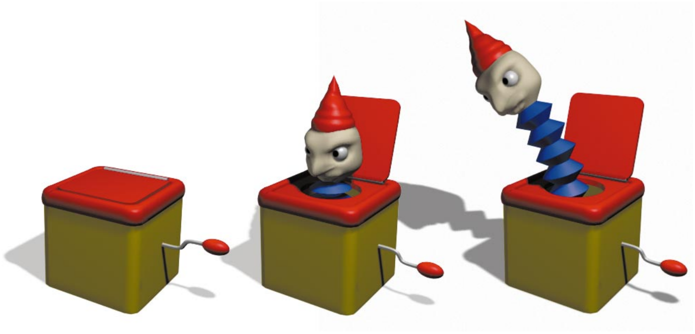


Head with eyes and hat

9 Animation

In the last two lessons, you modeled various pieces of the Jack-in-the-box. You can now set keyframes on the various parts to bring Jack to life. To start, you will build some skeleton joints that will help you control Jack. You will then add inverse kinematics to animate Jack's release.

In this lesson, you will also use Maya's character definition tools that will help you create a pose-based animation that lets you quickly block out the motion.



The Jack in the box

In this lesson you will learn the following:

- How to set up skeleton joints
- How to bind geometry onto joints
- How to set up a Spline IK solution
- How to create a new attribute
- How to constrain an object to a joint
- How to set up a character
- How to set keys on characters

Adding skeleton joints

Skeleton joints offer a great tool for controlling how a character moves. By moving the skeleton, you will move the surface.

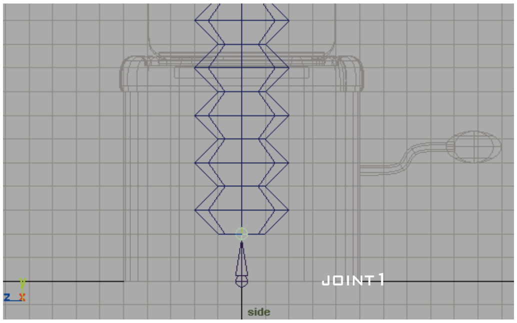
1 Prepare your scene

- **Open** the scene file that you saved in the last lesson.
- Place the head and the body surfaces on a new layer. Call the new layer *Jack_layer*.
- Select **Template** from the *box* layer's pop-up menu.

  Now you are focused on the Jack character.

2 Draw two skeleton joints

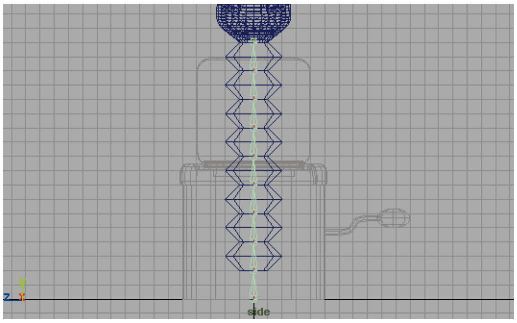
- Select **Display → Joint Size → 25%**.
- Go to the **Animation** menu set.
- Select **Skeleton → Joint Tool**.
- In a side view, press the **x** key to use grid snapping then click near the ground to place the first joint.
- Click two grid units above the first joint to place the second joint.



First Joint

3 Draw 8 more joints

- Use grid snap to place several more joints at two grid intervals until you reach the head.
- Press the **Enter** key to complete the skeleton.



Second Joint

IK Spline Handle

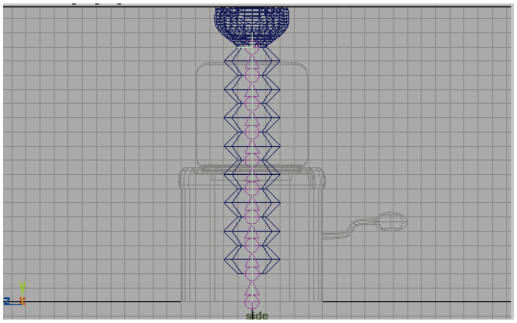
Later you will want to control Jack's head so that it sways back and forth after emerging from the box. You will use an Inverse Kinematics solution called IK Spline.

An IK Spline handle uses a curve to control the shape of the skeleton. As you move the curve, the Inverse Kinematics solver rotates the joints to match.

1 Create IK spline for the joints

- Select **Skeleton → IK Spline Handle Tool**.
- Click on the base of the skeleton when prompted for the *start joint for handle*.
- Click on the end of the skeleton when prompted for the *end joint or end effector*.

This adds an IK handle to the skeleton and a curve for controlling the joints.



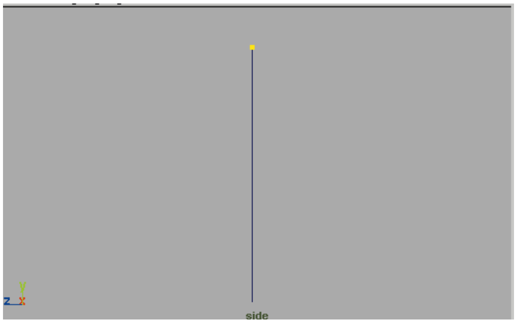
Spline IK handle

2 Add a point on Curve Deformer

You now have a single span curve that can be used to control the joints. You could edit the CVs on the curve, but instead you will add a Point on curve deformer for more high level control.

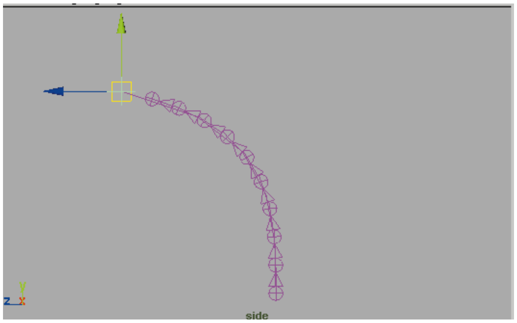
- From the side view panel, select **Show** → **None, Show** → **Grid** then **Show** → **Curves**.
- Click on the curve with your right mouse button.
- Select **Curve Point** from the pop-up marking menu.

  Click and hold on the curve then drag up to the top. This is your selected curve point.



Spline IK handle

- Select **Deform** → **Point on Curve**.
- Select **Show** → **Locators** and **Show** → **Joints.** Now you can see a locator at the top of the IK spline that will be used to control the curve.
- Select **Modify** → **Center Pivot**.
- **Move** the *Locator* to control the Spline.



Testing the point deformer

3 Skinning the neck surface

- Press **z** to undo the move.
- Select **Show** → **All**.
- With the Locator still selected, select **Display** → **Object Components** → **Selection Handles**.

This will make it easier to select the locator later, since these handles have a higher selection priority than other objects.

Skinning

To make the body surface move with the skeleton joints, it needs to be bound to the joints using Maya's skinning tools.

1 Skin the body surface.

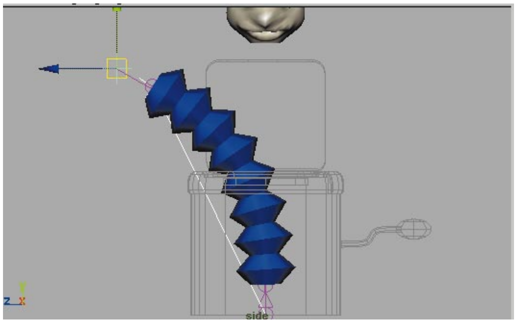
- **Select** the root joint of the skeleton.
- Press the **Shift** key and select the body surface.
- Select **Skin** → **Bind Skin** → **Smooth Bind**.

2 Testing the results

- Click-drag over the area where the locator is to select it.

  Since you added the selection handle, the *Locator* is selected over all the other objects when you click-drag to select.
- **Move** the *Locator* to test the bind.

  The smooth bind is very effective at giving good results very quickly. This bind is exactly what is needed.



Testing the bind

Set up the joint scaling

In order for Jack to pop out of the box, you need to set up the body so that it can extend out of the box or pull back into it.

1 Add an attribute to the locator

To control the extending of the body, you will add an attribute to the locator. This will give you a single attribute that will be linked to the joints using Set Driven key.

- With the *Locator* selected, select **Modify** → **Add Attribute...**
- In the Add Attribute window, set the following:

  Attribute Name to *extend*;

  Data Type to **Float**;

  Minimum to **0**;

  Maximum to **10**;

  Default to **10**.
- Click **OK**.

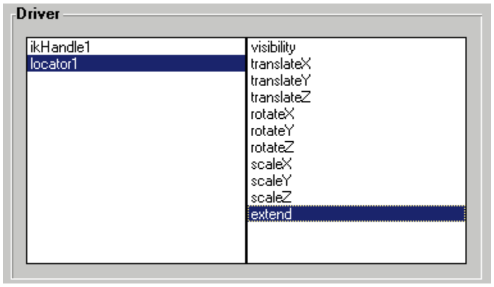
  In the Channel box, you will see that a new attribute named *Extend* has been added to the list. You will use this to control Jack moving in and out of the box.

2 Load the Set driven key window

Since the extending of the skeleton joints requires the scaling of several nodes, you will use Set Driven key so that editing the *Locator's* **extend** attribute does all the work.

- Select **Animate** → **Set Driven Key** → **Set** - ❐.
- In this window, click on the **Load Driver** button.

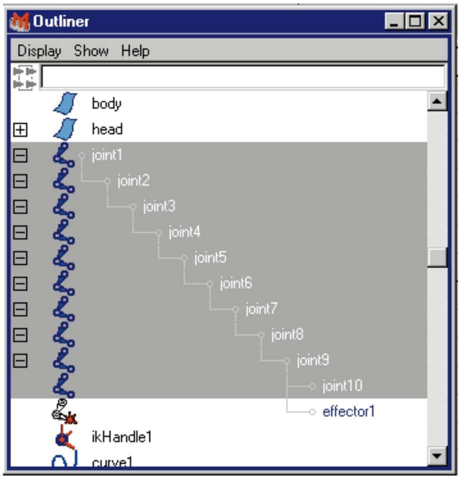
- Click on **Locator** in the left hand column.
- Click on **extend** in the right hand column.



Loading the driver attribute

3 Load all the joints as the driven

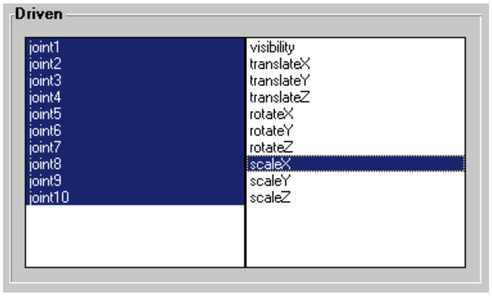
- Select **Window → Outliner**.
- Press the **Shift** key and click on the square box next to *joint1*. This will expand the hierarchy so that you can see all the joints.
- Press the **Shift** key and select all the joints in the skeleton.



Outliner

- Click **Load Driven** in the Set Driven Key window.
- In the left hand column, select all of the **joints**.
- In the right handle column select **scaleX**.

  By default, the joints' X-axes are pointing along the joints. For this reason this is the axis that needs to be modified.



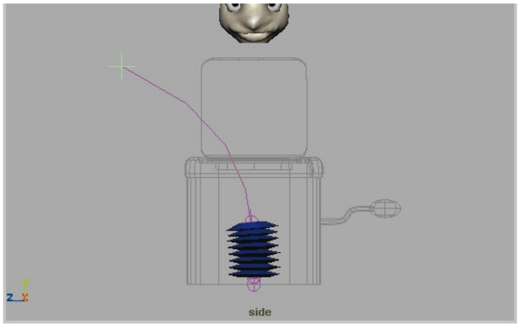
Loading the driven attributes

4 Key the start position

- Click on the **Key** button so that when the **extend** attribute is set to 10, the joints will be scaled to 1 in X.

5 Key the end position

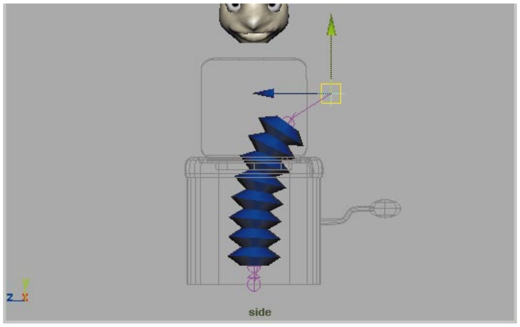
- **Select** the *Locator*.
- Set the **extend** attribute to **0**.
- Re-select all the joints in the Outliner.
- In the Channel box, set **ScaleX** to around **0.15**. This squashes the body into the box.
- Click on the **Key** button.



Setting a second key

6 Test the results

- **Select** the *Locator*.
- Click on the **Extend** name in the Channel box.
- With your middle mouse button, click-drag left to right to adjust the body surface.
- **Move** the Locator to see how the extending action works with the spline curve.



Testing the Expand attribute

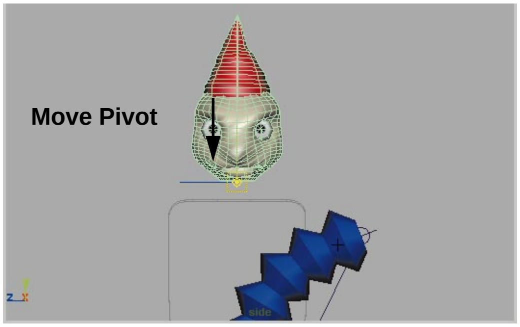
Constraining the head

You now want the head to move with the skeleton, but you don't want it to squash with the body. Therefore instead of binding the head into the skeleton, you will constrain the head's position and orientation to match the skeleton's top joint.

1 Edit the head's pivot location

- **Select** the *head* surface.

  This will also select the eyes and hat which are children of the head.
- Select **Modify → Center pivot**.
- Select the **Move** tool.
- Press the **Insert** key to go into edit mode.
- **Move** the head's pivot to the base of head.
- Press the **Insert** key to go back to object mode.

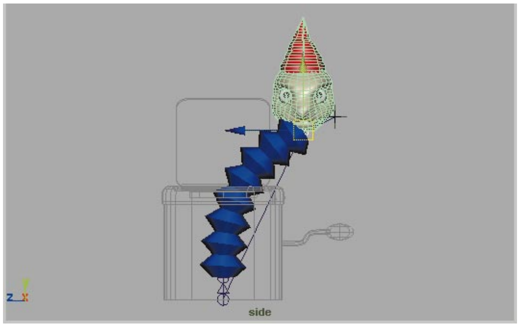


Setting the pivot location

2 Point constrain the head

- **Select** last *joint*. You may want to use the Outliner to help you.
- Press the **Shift** key and select the *head* surface.
- Select **Constrain → Point**.

  Now the head sits in the same position as the top joint. Right now it isn't oriented properly.



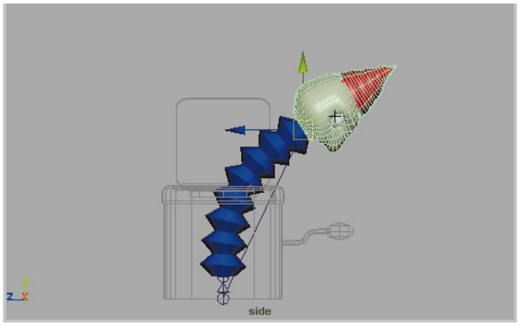
Point constraint

3 Orient constrain the head

- With the joint and the head still selected, select **Constrain → Orient**.

  Now the head is oriented to match the joint. One problem is that it has been rotated sideways. This is because the head's local axis and the joint's local axis work slightly differently.

  Every node has a local axis that determines its orientation in space. You can view and edit the axis to correct the rotation of the head.

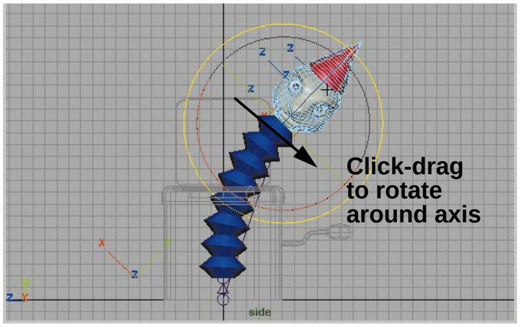


Orientation constraint

4 Rotate the local rotation axis

- **Select** the head surface.
- Press **F8** to go into component mode.
- Click on the **Points** selection mask button to turn it off then click on the **Miscellaneous** button to turn it on.

  This selection mask option displays the local rotation axes for all the parts of the head.
- Select the axis near Jack's neck.
- Select the **Rotate** tool.
- Click-drag on the Y Rotate manipulator until the head is facing forward. If you rotate the manipulator handle to the right, the head will rotate to the left.



Editing the local rotation axis

- Click on the **Miscellaneous** selection mask button to turn it off then click on the **Points** button to turn it on.
- Press **F8** to go back to object mode.

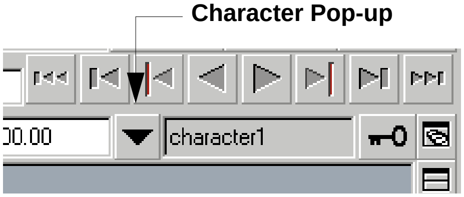
Set up a character

Before animating the whole scene, you are going to create several Maya character definitions. A Character definition is a collection of attributes that you need to animate for your chosen character.

1 Create a new character

- Select the *Locator*.
- Select **Character → Create**.
- Select **Character Sets** from the Character pop-up menu found just under the playback controls in the bottom right of the workspace.

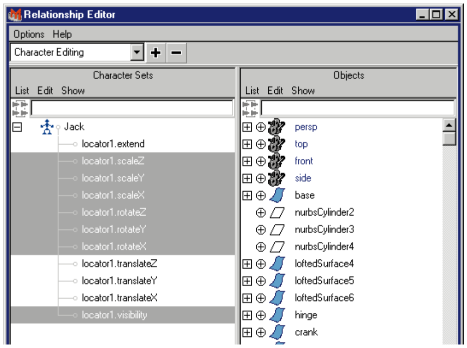
Character Pop-up



Testing the bind

- In the Relationship editor, double click on name character1 and change it to *Jack*.
- Click on the square icon next to the character to expand the list of associated attributes.

  All of the Locator's keyable attributes have been added to the character.
- Select everything except **extend** and **Translate X, Y** and **Z.**



Removing attributes

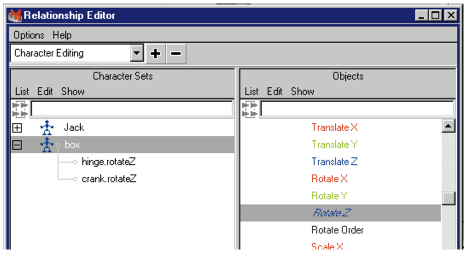
- Select **Edit → Remove Highlighted from character**.

  Now you have the character definition restricted to the remaining attributes.

2 Create a second character

To animate the parts of the box, you will create another character that uses other attributes that you would associate with the box.

- With nothing selected, select **Character → Create**.
- Rename this character *Box*.
- In the left hand list of the Relationship editor, select the new character.
- In right hand list, find the *hinge* node.
- Click on the round icon to expand the node and view the available attributes.
- Select **RotateZ** in this list so that this attribute is added to the character.
- Repeat these steps to also add the *crank* node's **RotateZ** to the character.

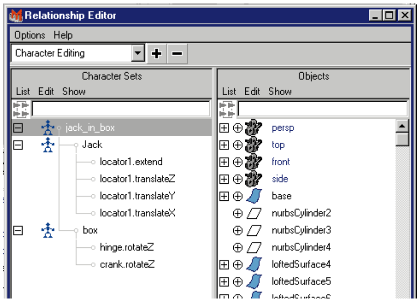


Adding attributes

3 Combine the characters.

You can now create another character and use it to combine the *Jack* character and the *Box* character.

- With nothing selected, select **Character → Create**.
- Rename this character *jack_in_box*.
- In the left hand list of the Relationship editor, select the new character.
- In the right hand list scroll down until you find the *Jack* character then select it.
- Repeat for the *box* character.
- Now click on the square icon next to *jack_in_box* and you will find that the first two characters are now sub-characters of the first.



Expanded character

Tip: This method can also be used to create definitions for a character's body parts such as arms and legs.

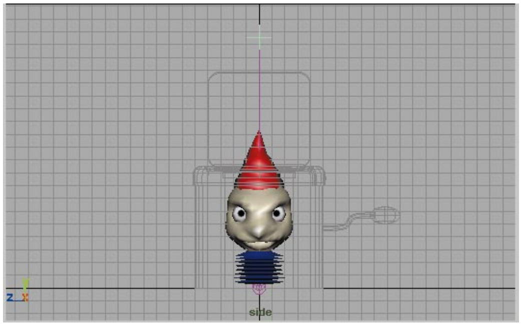
Setting up for animating

You are almost ready for animating the scene. Before you start, you need to get Jack and the box into their starting positions.

1 Position the Locator

To set up Jack, you need to set the locator's extend attribute so that Jack is hiding in the box.

- **Move** the *Locator* so that it is sitting directly above the box.
- Set the *Locator's* **extend** attribute to **0**.
- You may notice that Jack's hat is sticking up past the top of the box. You will need to scale the head down when it is in the box.

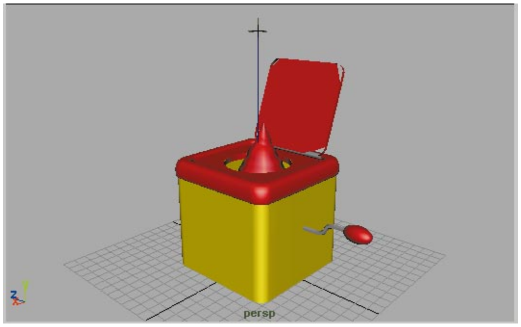


Locator positioning

2 Untemplate the box layer

- From the *box_layer's* pop-up menu, select **Standard**.

  Now you can see all of the parts of the model.



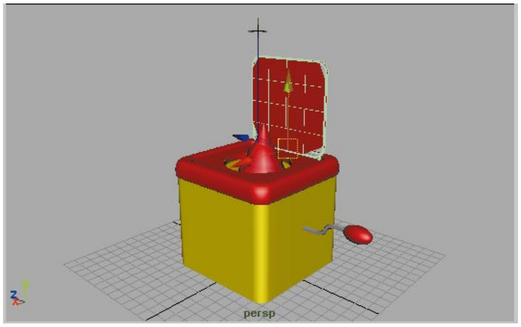
Open position

3 Set head to scale with the hinge

To set up the scaling of Jack's head, you will use Set Driven key. This will allow you to control the scaling of the head by the rotation of the hinge.

- Select **Animate** → **Set Driven Key** → **Set** - ❐.
- Select the *hinge* then click on the **Load Driver** button.
- Click on *rotate Z* in the right hand column.
- **Select** the *hat* surface then press the **up arrow** to navigate the hierarchy to the *head*.

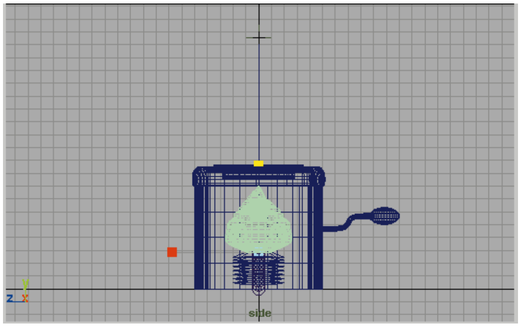
  You can use the up and down arrows to navigate between children and parent nodes in a hierarchy.
- Click **Load Driven**.
- Click on *scale Y* in the right hand column.
- **Rotate** the hinge to about **90** degrees around the Z axis.
- Click on the **Key** button so that when the lid is all the way open the head will be scaled to 1 in Y.



Open position

4 Key the end position

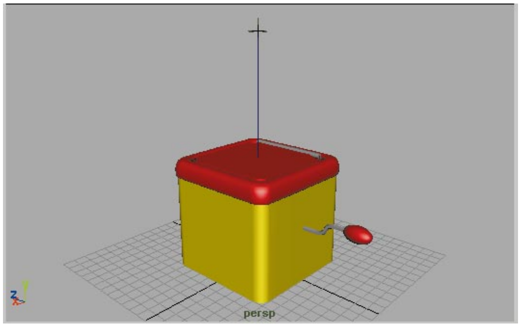
- **Select** the *hinge*. and set **Rotate Y** to **0**.
- Select the *head* and set its **Scale Y** so that the hat is sitting inside the box.
- Click on the **Key** button.



Closed position

5 Set up the box

- Set the **Rotate Z** of both the *hinge* and the *crank* to **0**.
- You are now ready to animate the Jack-in-the-box.



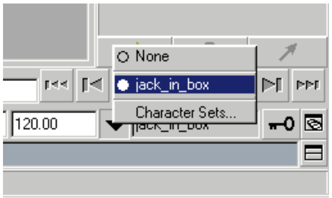
Starting position for box

Animate the Jack in the box

Because you set up the Jack-in-the-box as a character, you can quickly flesh out your animation using a pose-based technique. This means that you set up a series of poses for the character then set keys for all the attributes.

1 Set up for the animation

- In the time range, set the **Start frame** and **Playback start frame** to 1 and the **End frame** and **Playback end frame** to 120.
- In the Character menu, select **jack_in_box**. This does not highlight the character in the modeling view. It only indicates it as the active character.



Character menu

Now when you set keys using the **s** key, all the character's attributes will be keyed. When working with a character, it is important to remember that you can set keys on the active character even if the related nodes are not selected in the modeling views.

2 Set your tangent defaults

You will edit your tangent defaults so that all keys are created with **Flat** tangents. This tangent option gives a good starting point for the pose-base animation being used here.

- Select **Options** → **General Preferences**.
- Click on the **Animation** tab then scroll down to the **Keys** section.
- Set the following:

  Default In Tangent to **Flat**;

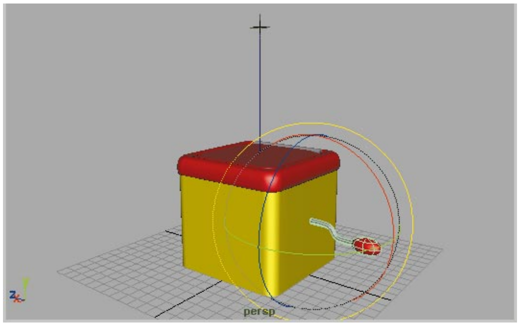
  Default Out Tangent to **Flat**.

3 Set an initial key

- Move the Time slider to frame 1.
- Press the **s** key to set keys.

4 Create a pose at frame 60

- Move the Time slider to frame 60.
- Select the *crank*.
- Set its **Rotate Z** to **900**.
- With your middle mouse button, click in a modeling view then press the **s** key to set keys.



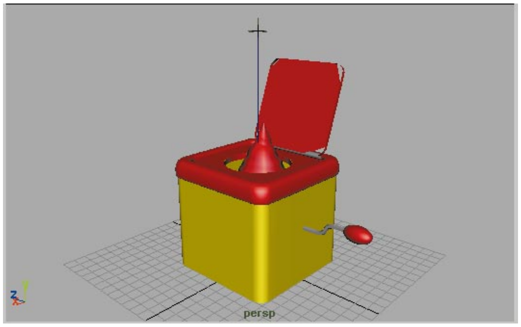
Rotated crank

Note: Even though the crank is selected in the modeling view, the keys are being set for all the attributes in the *jack_in_box* character.

5 Create a pose at frame 70

- Select the *hinge*.
- Set **Rotate Z** to around **110**.
- With your middle mouse button, click in a modeling view then press the **s** key to set keys.

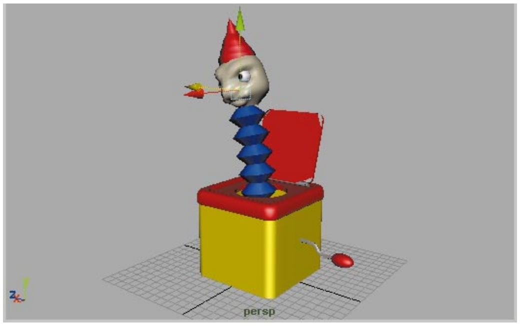
  Now the lid is open and the head has scaled back to its normal size.



New Pose at frame 70

6 Create a pose at frame 80

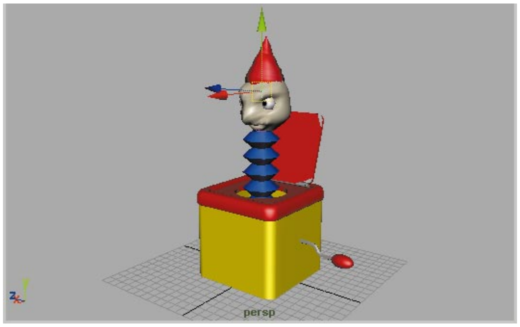
- Select *Locator* and set **Extend** to **10**.
- **Move** the *Locator* to the left.
- Set another key.



New Pose at frame 80

7 Create a pose at frame 85

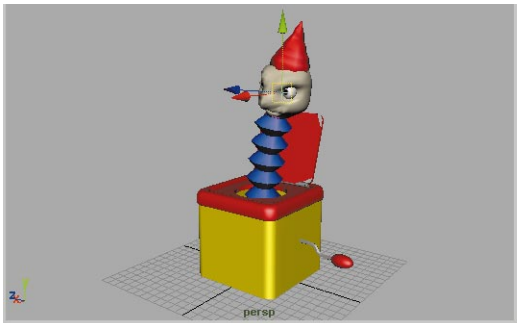
- Select *Locator* and set **Extend** to **7**.
- **Move** the *Locator* to the middle.
- Set another key.



New Pose at frame 85

8 Create a pose at frame 90

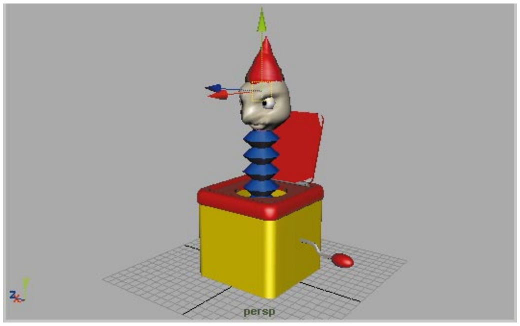
- Select *Locator* and set **Extend** to **10**.
- **Move** the *Locator* to the right.
- Set another key.



New Pose at frame 90

9 Create a pose at frame 95

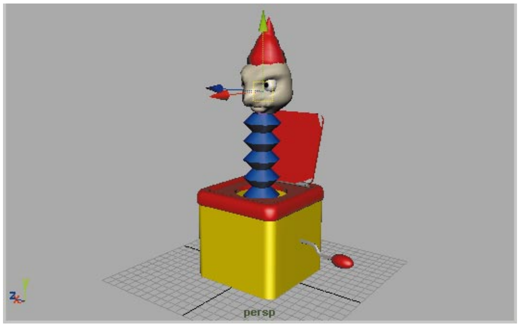
- Select *Locator* and set **Extend** to **6**.
- **Move** the *Locator* to the middle.
- Set another key.



New Pose at frame 95

10 Create a pose at frame 100

- Select *Locator* and set **Extend** to **10**.
- **Move** the *Locator* to the left.
- Set another key.

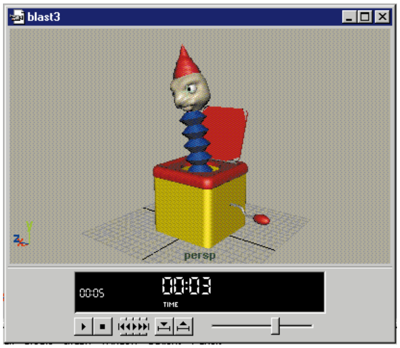


New Pose at frame 100

11 Preview the results

- Make sure that your perspective panel is active.
- Select **Window → Playblast...**

This creates a quick preview animation that is good for reviewing the look of an animation.

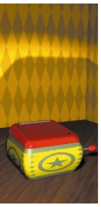


Playblast window

For the Jack-in-the-box, you will see that the motion is a little mechanical. While the pose-based animation has not given you a finished product, it did offer a quick way of blocking out the action. In the next lesson, you will fix the look of the animation using several secondary motion techniques.

12 Save your work

Conclusion



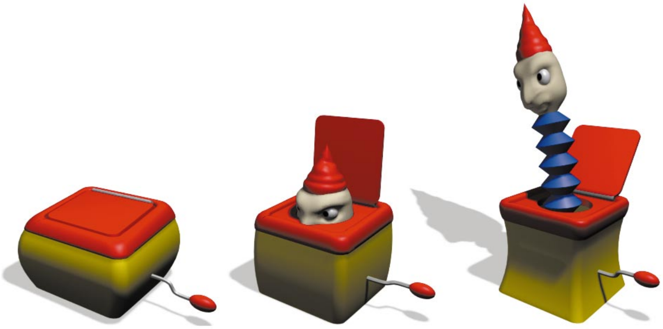
By creating a character definition for the Jack-in-the-box, you were able to quickly set up poses to get a quick look at the motion. The character definition lets you bring different attributes from a number of different nodes and put them in one easy to organize place.

In the next lesson, the character will be used to help you edit the animation curves. You will then add some deformers to your surfaces to create some anticipation and follow-through for the action.

10 Secondary Animation

This lesson is about adding some life to the mechanical motions generated by the pose-based animation of the last lesson. You are going to use traditional animation techniques such as anticipation, overlapping action and squash and stretch to enhance the quality of the motion.

In Maya, you have a lot of flexibility when it comes to refining your animations. You can edit the curves in the graph editor or you can add deformers to give life to a scene. By combining several of these techniques you will create a more effective animation.



The Jack in the box

In this lesson you will learn the following:

- How to create overlapping actions in the graph editor
- How to use breakdown keys.
- How to insert in-between keys
- How to use a lattice deformer
- How to lock an object to a deforming surface

OVERLAPPING ACTIONS

When you look at the animation of the Jack-in-the-box, each action follows directly after the last. The crank rotates. As it stops the lid opens. After the lid opens, Jack emerges. This breaks down the flow of the animation by giving distinct start and stop points.

You are now going to adjust the animation curves belonging to these actions so that each one flows into the other. You will continue working with the file you completed in the last lesson.

Extending the crank rotation

The first place where you can overlap the action is the crank rotation. Currently it stops just as the lid is being opened. You will extend the crank rotation to overlap the lid.

1 Open a graph editor panel

- Set up a panel layout where you have a perspective view, a Graph editor and any other panel type that you want.

2 View box curves in the graph editor

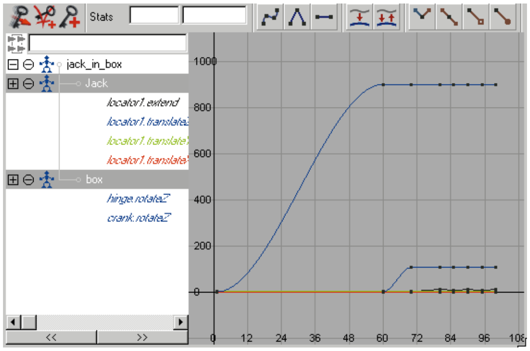
- From the character menu below the playback controls, select the *jack_in_box* character.

  The character now appears in the Graph editor. This is another advantage of using the character definitions. You can have the character loaded in the Graph editor even if no objects are selected in the modeling views.

- Click on the square plus sign next to jack_in_box to expand the character.

  You now see the two sub-characters and all of their attributes.

- Press the **Ctrl** key and click on the *Jack* and the *box* characters.
- From the Graph editor, select **View → Frame All**.



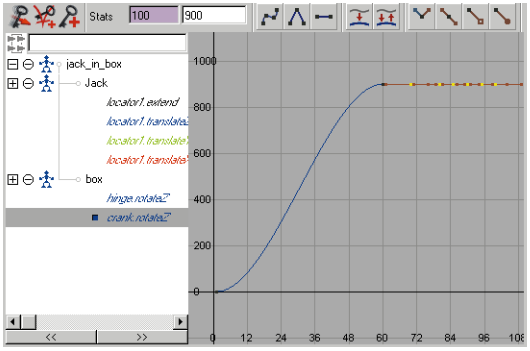
Graph editor

3 Delete extra keys on crank curve

- Click on the *crank.rotateZ* channel in the Graph editor.

  Now it is the only visible channel.

- **Select** the keys at the end of the curve. Leave only the first two unselected.

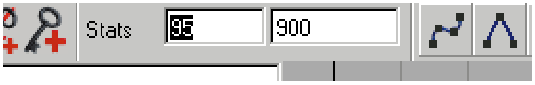


Selected keys

- Press the **Backspace** key to remove these keys.

4 Move the last key

- Select the last key on the curve.
- In the **Stats** section of the Graph editor, change the **frame** value to **95**.

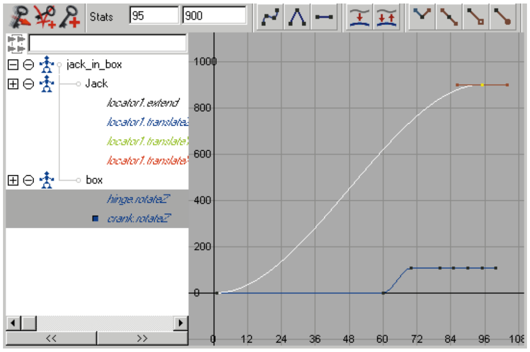


Graph editor stats

- Press the **Shift** key and click on the *hinge.rotateZ* channel in the Graph editor.

  You can see that the rotation of the crank is now overlapping the opening of the lid in time.

- Playback the results in the perspective panel to see the overlap.



Viewing overlap in the Graph editor

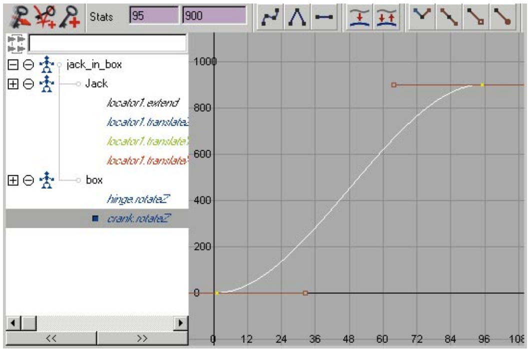
Editing Tangent Weights

Since the *crank.rotateZ* channel is only described by two keys, the quality of its motion will be defined by what is happening between the keys. The Flat tangents set in the last lesson ensure that the crank accelerates into motion at the beginning and decelerates to a stationary position at the end.

To control these two actions, you need to edit the weighting of the existing tangents so that you can either lengthen or shorten the action at the end points.

1 Convert the keys to weighted keys

- Click on the *crank.rotateZ* channel in the Graph editor so that it is again the only visible channel.
- Select both keys in the Graph editor.
- From the Graph editor, select **Curves → Weighted Tangents**.
- Next, click on the **Free tangent weight** button. This lets you edit the tangent weight on one side of a key without editing the tangent weight on the other.

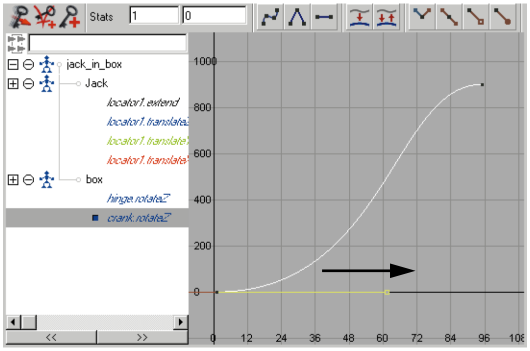


Added tangent weight

2 Adjust the tangent weight

- **Select** the first key on its own.
- **Select** the Tangent weight handle.
- Select the **Move** tool.
- Press the **Shift** key and click-drag to the right.

This increases the weighting.

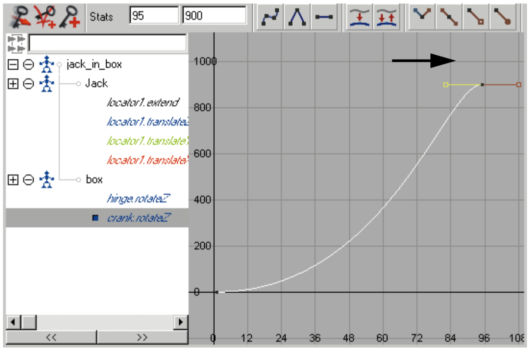


Added tangent weight

3 Adjust the end key tangents

- **Select** the second key on its own.
- **Select** the Tangent weight handle.
- Press the **Shift** key and click-drag to the right.

  This decreases the weighting.



Added tangent weight

- Playback the results.

  Now the handle takes longer to start up but stops more abruptly.

Overlapping Jack's extending

Another place where you could use some overlap is where Jack begins to extend up from the box. Right now, Jack doesn't emerge from the box until after the lid is actually opened. This is a result of the pose-based animation that was used. By deleting just one key, you can get some overlap here.

1 Select the extend channel

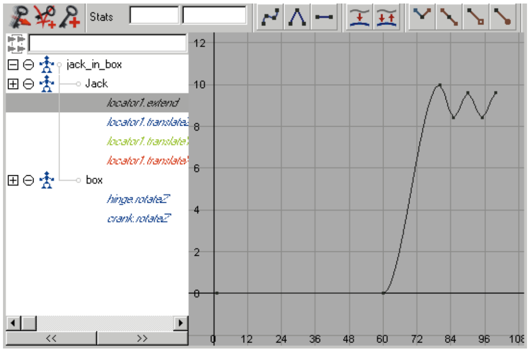
- In the Channel box, click on the channel labelled *locator.extend*.
- Select **View** → **Frame All**.

2 Select the extra key

- **Select** the key at frame 70.

  This is the key that matches the point where the lid is fully opened.
- Press the **Backspace** key to delete the key.
- Playback the results.

  Now Jack begins to emerge as the lid is opening.



Extend animation curve

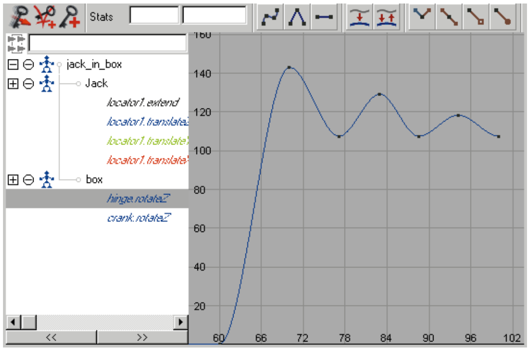
Add follow-through to the lid

When the hinge rotates, it goes from a closed position to an open position directly. To help emphasize the force of the lid's opening, you will now create some follow-through when the lid is fully opened. Rather than stopping it will sort of bounce back and forth to suggest a reverberation from the opening.

1 Move the extra keys

- **Select** the *hinge.rotateZ* attribute in the Channel box.
- From the Graph editor, select **View → Frame All.**

  This curve has a number of extra keys after the point where the lid opens. Rather than deleting these, you will modify them to create the back and forth motion.
- **Select** the last six keys on the curve.
- From the Graph editor, select **View → Frame Selected.**
- **Select** and **Move** the frames as shown below. Notice how the key at frame 70 has had its value increased to about 140 degrees. The goal is to have the lid open extra wide at frame 70 then oscillate back and forth for the next few frames:



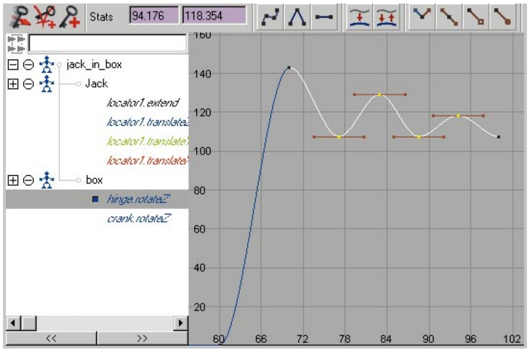
Updated frames

- Playback the results.

2 Convert keys to breakdown keys

Now that the final 6 keys are working correctly, you want to work with them as a grouped action. By converting the middle keys to Breakdown keys, you can control them all using the last key.

- **Select** the four keys that sit between the key at frame 72 and the end key.
- From the Graph editor, select **Keys → Convert to Breakdown.**

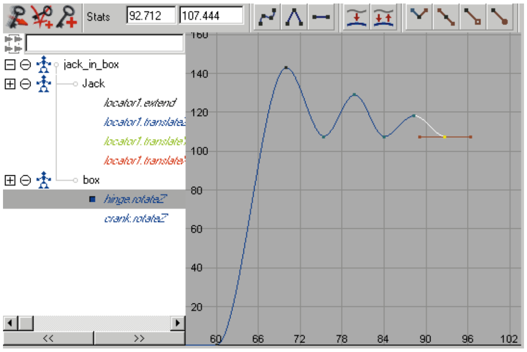


Selected keys

3 Test the breakdown keys

A breakdown key is a special key in Maya that maintains a relationship to the keys around it. If they move, then the breakdown key is adjusted to suit.

- **Select** the end key.
- Press the **Shift** key and with your middle mouse button, **Move** the key left and right to move it in time.
- You can see that now the breakdown keys maintain an appropriate relationship to the keys around it.



Editing the end key

- Playback the scene. If you are happy with the look of the lid then move on. If not, then continue to edit the keys until you are happy.

ANTICIPATION

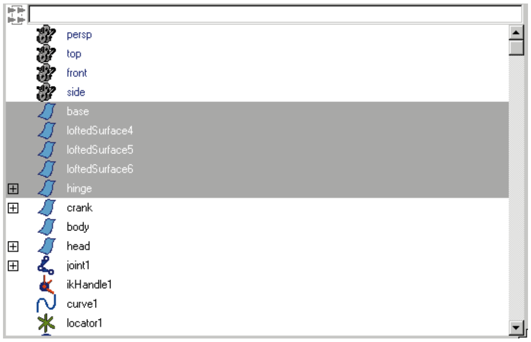
Another important animation concept is anticipation. If you playback the Jack-in-the-box scene, you will notice that your audience is given no clue that Jack is going to emerge. One way to offers some anticipation would be to add a cartoon-like squashing of the box. As the crank begins to wind, the box begins to bulge until at the last minute it bursts with Jack emerging.

Add lattice to box

To create the bulging of the box, you will use a lattice deformer. This deformer type allows you to reshape one or more surfaces using a special frame. By working with the Lattice points, you can deform the look of the surface.

1 Delete history

- Select **Window → Outliner**.
- **Select** the base surface and the three lofted surfaces and the hinge.



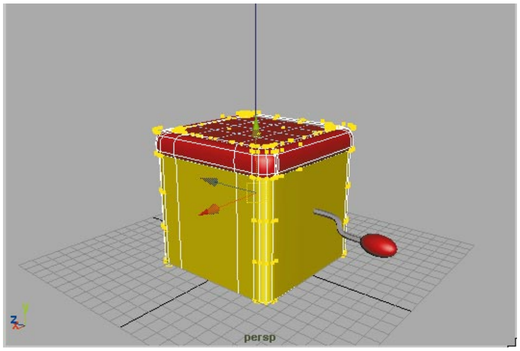
Selected surfaces

- Select **Edit → Delete by Type → History**.

  This is to make sure that the Lattice works correctly on these surfaces.

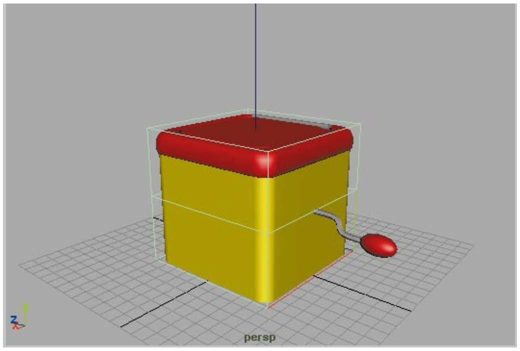
2 Add the Lattice

- Press **F8** to go into component mode.
- **Select** all the CVs for all the surfaces.



Selected control vertices

- Select **Deform → Create Lattice**.
- In the Channel box, set the **T divisions** to **3**.



Box Lattice

Animate the box deformations

To animate the bulging of the box, you need to animate the lattice points themselves.

1 Set keys for frame 1

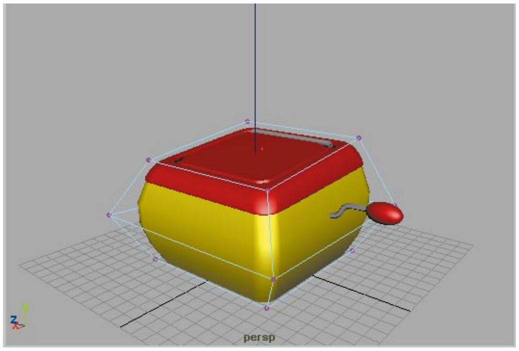
- Go to frame 1.
- From the **Character** menu, select **Set Current Character → none**.

  Setting this to none is *very important* to make sure that any keys you set do not get set on other character nodes.
- Click on the ground to deselect the Lattice then click on the Lattice to re-select it.

  This allows the Lattice points to become visible.
- **Select** all the Lattice points.
- Press **s** to set keys for the points.

2 Set keys for frame 50

- Go to frame 50.
- Edit the Lattice points to get the following shape. You will need to pick Lattice points either one at a time or as a group then either **Move** or **Scale** them. The goal is for the box to appear squashed with its front end tilted slightly lower than the back end:

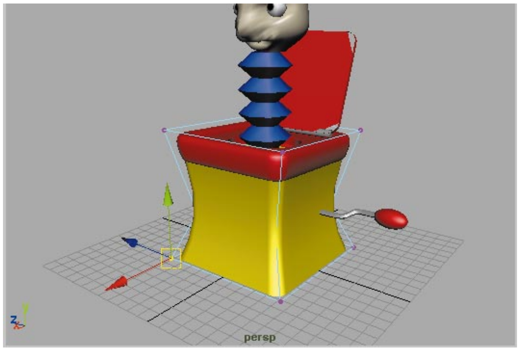


Box lattice at frame 50

- **Select** all the Lattice points.
- Press **s** to set keys for the points.

3 Set keys for frame 72

- Go to frame 72.
- Edit the Lattice points to get the following shape:

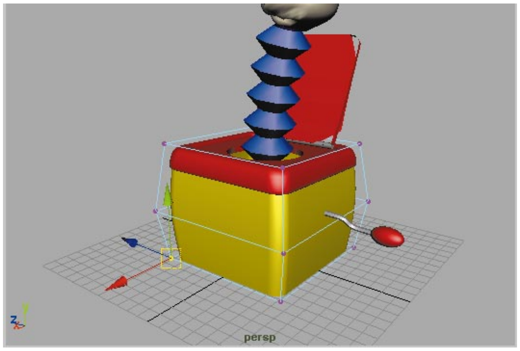


Box lattice at frame 72

- **Select** all the Lattice points.
- Press **s** to set keys for the points.

4 Set keys for frame 82

- Go to frame 82.
- Edit the Lattice points to get the following shape:

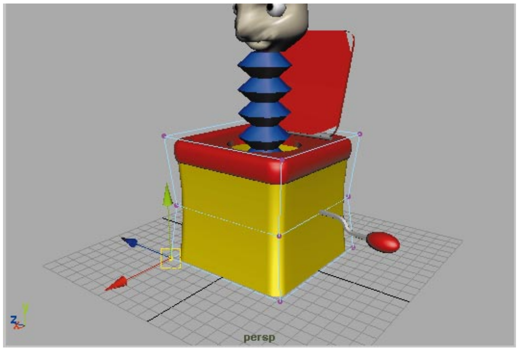


Box lattice at frame 82

- **Select** all the Lattice points.
- Press **s** to set keys for the points.

5 Set keys for frame 95

- Go to frame 95.
- Edit the Lattice points to get the following shape:



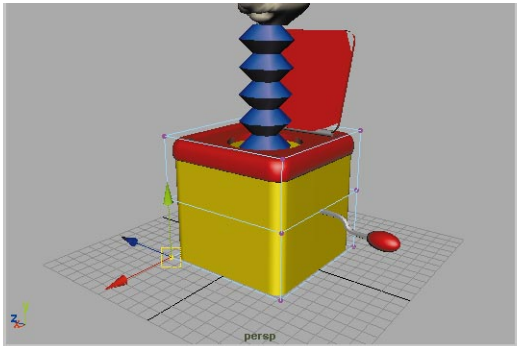
Box lattice at frame 95

- **Select** all the Lattice points.
- Press **s** to set keys for the points.

6 Set keys for frame 100

- Go to frame 1.
- With your middle mouse button, drag the current time indicator in the time slider to frame 100.

  By using the middle mouse button, the time was updated but the objects stay in their frame 1 positions.
- **Select** all the Lattice points.
- Press **s** to set keys for the points.



Box lattice at frame 100

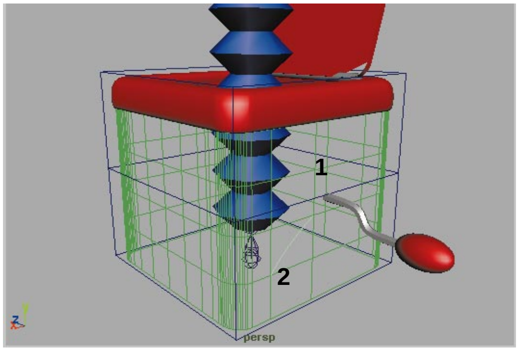
Constrain crank to box surface

As you playback the scene, you will notice that the base of the box bulges but the crank remains stationary. You can use a single frame motion path to add the crank to the bulging action.

1 Draw a curve on surface

- Press **F8** to go back into object selection mode.
- Select the *base* surface of the box.
- Select **Modify** → **Make Live**.
- Hide the grid for the perspective view panel.

  Now the base surface is the working surface. Its isoparm lines will be the grid on which you will draw.
- Select **Create** → **EP Curve Tool**.
- Press the **x** key and click on the isoparm crossing at the center of the side face.
- Click a second point somewhere else on the surface.
- Press the **Enter** key to complete the curve.



Crank attached to surface

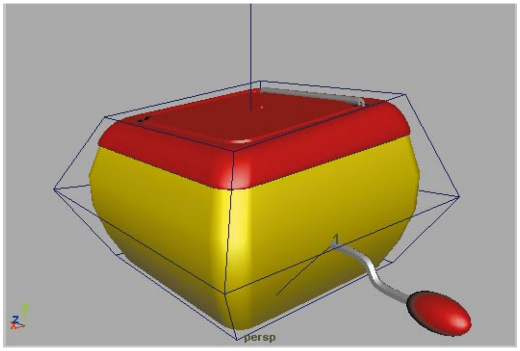
- Select **Modify** → **Make Live** to make the ground grid active again.

2 Attach Crank to path

- **Select** the *crank* then press the **Shift** key and select the curve on surface.
- Select **Animate** → **Paths** → **Attach to Path** - ❐.
- In the option window, set the following:

  Time Range to **Start**.
- Click **Attach** then **Close**.
- Playback the results.

  Now the crank is locked to the curve on surface and follows the surface during its deformations.



Crank attached to surface

SQUASH AND STRETCH

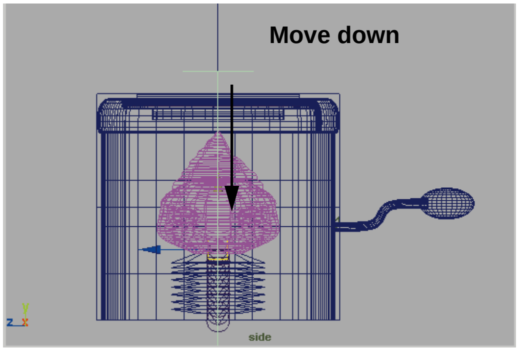
In the bouncing ball example, you learned about squash and stretch. This principle will also be useful when Jack's head reaches the peak of his extension. Right now you notice that the head reaches the peak with no visible reaction. By adding a squash deformer to the head, you will be able to stretch the head at these points to give the feeling that the head is following through with the action.

Add squash deformer to head

To make the deformer work with the head, you will have to add it to the appropriate shapes then parent it into the hierarchy.

1 Add the deformer

- Select the *head* surface. This also selects the eyes and the hat.
- Select **Deform → Create Nonlinear → Squash**.
- Move the *squashHandle* down to the base of the head.



Squash deformer

2 Parent the deform handle

- In the Outliner window, click with your middle mouse button on the *squashHandle* and drag it to the head node. This action parents these together.

Animate stretching of head

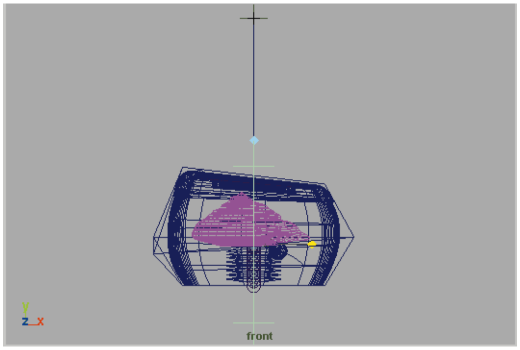
You can now set keys on the squash attribute to complete the animation.

1 Set a key for frame 50

Since the box is itself a little squashed at frame 50, you will start here so that you can squash the head enough to make sure that the hat is not poking through the lid.

- Go to frame 50.
- With the *squashHandle* selected, click on the squash input node in the Channel box.
- Select the **Show Manipulator** tool.
- Click-drag on the squash manipulator to squash the head down.

- In the Channel box, click on the squash input node's **Factor** attribute's name to highlight it.
- From the Channel box menus, select **Channels** → **Key Selected**.



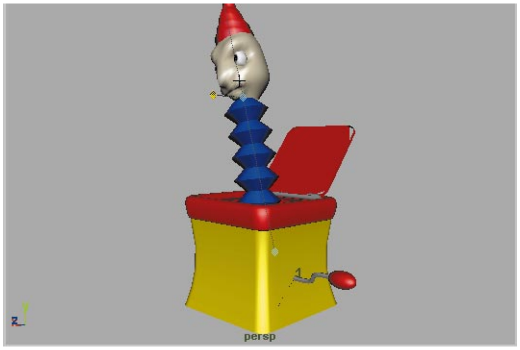
Squash setting

2 Set a key at frame 80

To set the remaining keys, you will use Auto keying.

- Turn on **Auto Key**.
- Go to frame 80.
- Use the *squashHandle* manipulators to stretch out Jack's head.

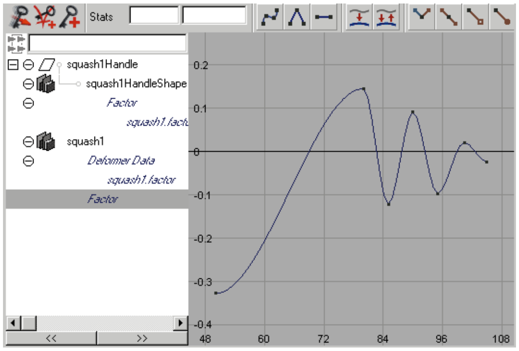
  A key is automatically set.



Squash at frame 80

3 Set more keys

- Now follow the animation and wherever Jack's head hits a peak in its bouncing, add some stretch, and wherever his head hits a low point, add some squash.
- Your final animation curve for the Factor attribute might look something like this.
- When you are finished turn **Auto Key** to off.



Squash factor curve

4 Save your work

Finishing touches

If you playback the animation at this point, you may be happy with some parts of the motion and unhappy with others. It is now up to you to complete the animation by tweaking the various keys to create the perfect animation.

Here are some examples of what you might want to do:

1 Make the head bounce for longer

Right now the head only bounces up and down for a short time. Extend these keys beyond the point where the box stops deforming. You may even want to add a few more bounces.

Remember that if you edit the timing of the Locator's **extend** attribute you must do the same for the squash's **factor** attribute since these work together.

2 Make the head flop around more

You may want to be more dramatic with where the Locator is positioned during the final bounces of the head. You could make these edits in the Graph editor or by setting new keys on top of old ones.

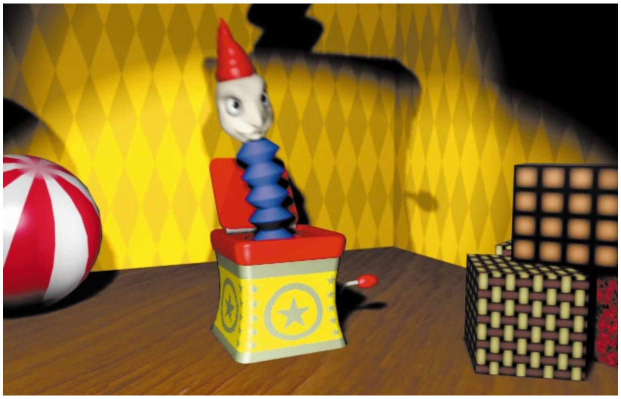
Conclusion

This lesson shows how time taken to add secondary animation to your work is very important to its success. Maya has many tools for adding these kinds of tweaks and adjustments and you should learn to use them effectively. Becoming an expert with the Graph editor is an important step to becoming a master animator.

11 Rendering

In this lesson, you will set up materials, textures and lighting for Jack. You will explore the make-up of a typical shading group including its material node and any texture maps.

This lesson will make extensive use of Maya's Interactive Photorealistic renderer (IPR). This tool allows you to create one rendering of the scene that can then be used to interactively update changes to the lighting and texturing of your scene. You will see how fast and intuitive it is to texture in Maya's IPR world.



The Jack in the box

In this lesson you will learn the following:

- How to set up and tweak multiple lights to a scene
- How to set up an Interactive Photorealistic Rendering
- How to add bump and specular maps
- How to make connections in the Hypershade panel
- How to texture map a deforming object
- How to create a texture reference object
- How to set up camera attributes

CREATE A SIMPLE SET

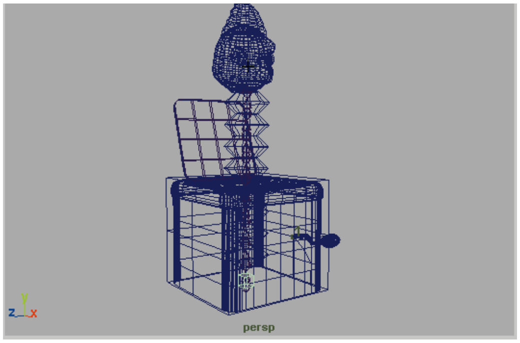
To create a backdrop for Jack, you will build a simple set with some props. These will all be built with primitives and will get their look from texture maps.

Create a three sided backdrop

You will create a cube then remove some of the faces to create a backdrop. Two walls and one floor surface is all you need to create a simple set for your action.

1 Create and position a primitive cube

- Select **Create → NURBS Primitives → Cube**.
- Set the *cube's* **Y translate** to **0.5**.

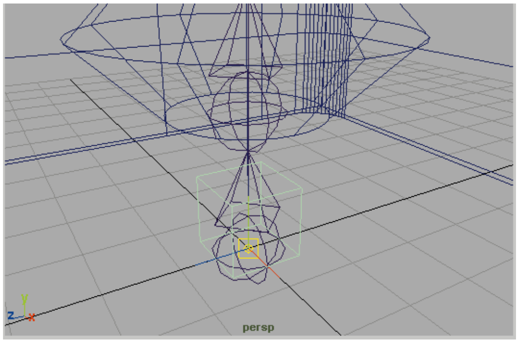


New cube

2 Set the cube's pivot

- Select the **Move** tool.
- Press the **Insert** key so that you can edit the cube's pivot point.
- Use grid snapping to move the cube's pivot point back to the origin.

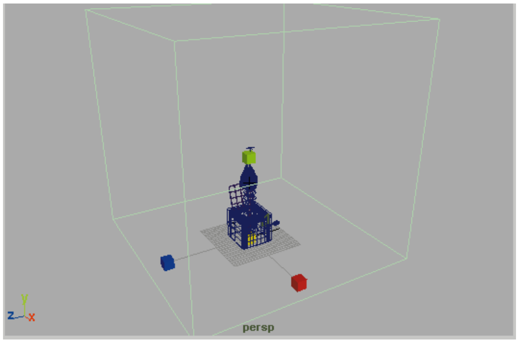
Now you will be able to scale the cube from its base.



Pivot location

3 Scale the cube to desired size

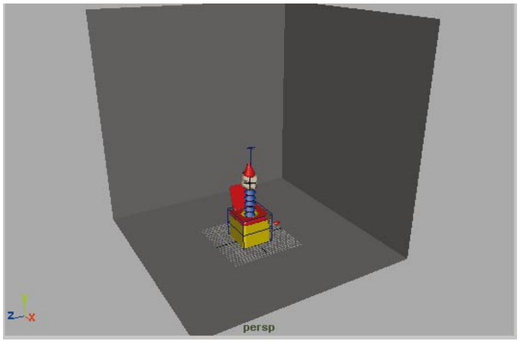
- Press the **Insert** key again.
- **Scale** the cube until it is big enough to cover a fairly large area round Jack.



Scaled cube

4 Remove the three unwanted faces

- Press **5** to turn on hardware shading.
- Select the top face and the two front faces and press the **Backspace** key.



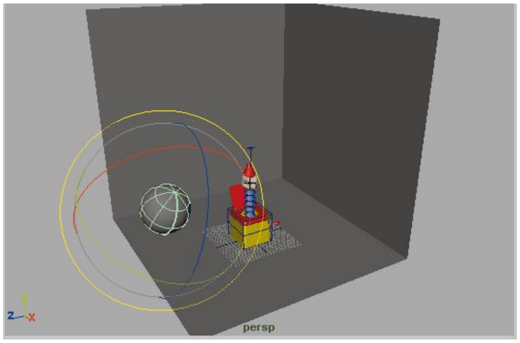
Three sides removed

Add some props

To add some extra detail to the scene, you will add some primitive shapes to act as props.

1 Create a ball

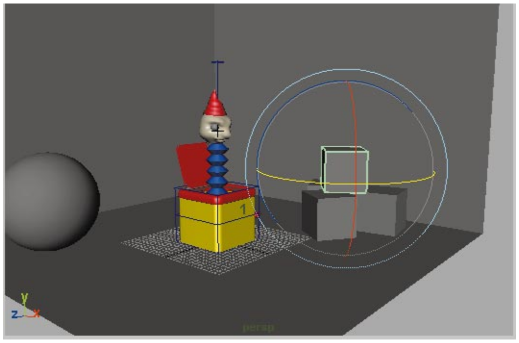
- Select **Create → NURBS Primitives → Sphere**.
- **Move** and **Scale** the sphere and place it to the left and behind the jack-in-the-box.
- **Rotate** it off its axis so that it looks more interesting when you texture it later.



Ball

2 Create some blocks

- Create another primitive cube.
- Position it to the right of the jack-in-the-box.
- **Duplicate** the box three times and create a stack of blocks.



Blocks

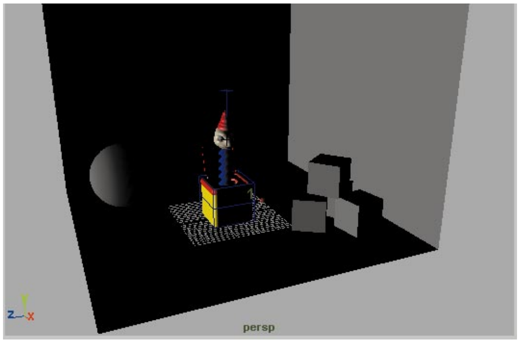
A Directional Light

Before you can start rendering your scene, you will need some light. The easiest light to set up is a directional light which works with parallel rays of light much like the sun.

1 Create a directional light

- From the **Rendering** menu set, select **Lights → Create Directional Light**.
- Click in your perspective panel with your middle mouse button then press the **7** key to turn on hardware lighting.

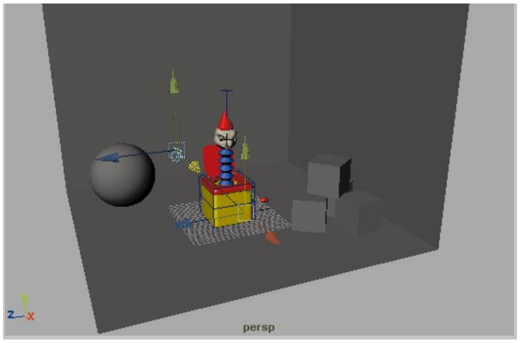
  At first, only the sides of your objects are illuminated because of the light's default direction.



Directional light

2 Position and Orient the light

- Select the **Show Manipulator** tool.
- Click drag on the light manipulator handle to move the light up and to the left.



Repositioned light

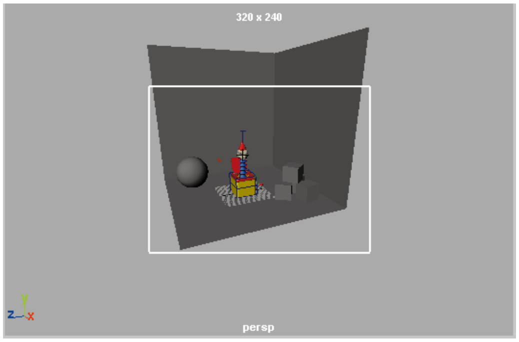
Camera setup

To see how you rendering will look when rendered, you can set up some display tools that will help guide your action.

1 Set up resolution gate

- From the perspective panel menus, select **View** → **Camera Settings** → **Resolution Gate**.

This sets up a bounding box to show you what will get rendered.



Resolution gate

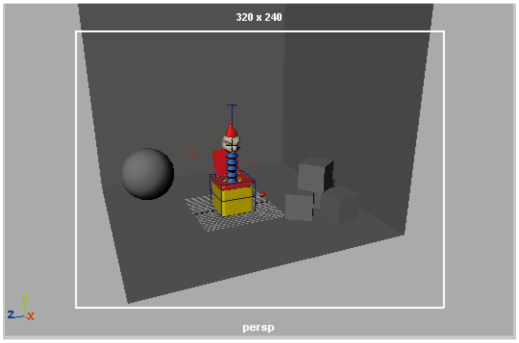
2 Adjust the camera's overscan

- From the perspective panel menus, select **View** → **Select Camera**.
- Press **Ctrl a** to open up the Attribute editor.
- In the Film Back section of the perspShape node, set the following:

  Film Fit to **Vertical**;

  Overscan to **1.2**.

Now you can see more of the scene.



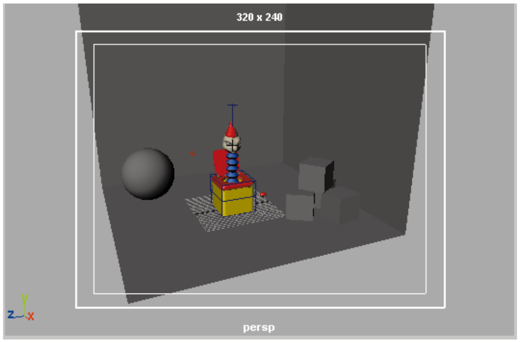
Resolution gate

Note: If your panel was more vertical then your Film Fit would have to be set to Horizontal.

3 Set up display guides

- From the perspective panel menus, select **View → Camera Settings → Safe Action**.

  This shows a second bounding box that defines the area that would be visible if you rendered to video.



Safe Action box

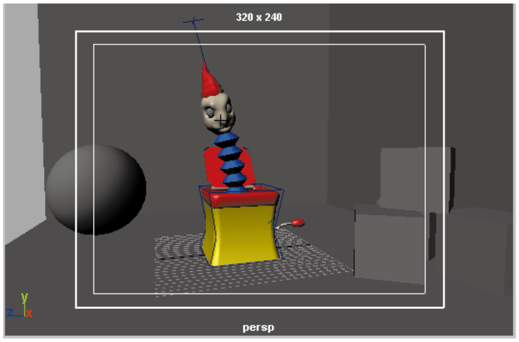
4 Set up a view for rendering

- From the perspective panel menus, select **View → Select Camera**.
- Press **Ctrl a** to open up the Attribute editor.
- In the **Camera attribute**s section of the *perspShape* node, set the following:

  Focal Length to **28**;

  This adjusts the Angle of view and lets you see a more wide angle view of the scene.

- Tumble. Track and Dolly into the view to get a nice view of the Jack-in-the-box extended. Scroll in the time slider to check your view at different times.



New view

IPR

You are now ready to render your scene. To give you access to interactive updating capabilities, you will set up an IPR (Interactive Photorealistic Rendering) rendering.

An IPR rendering creates a special image file that stores not only the pixel information about an image but also information about the surface normals, materials and objects associated with each of these pixels. This information is then updated as you make changes to your scene's shading.

1 Panel setup

- Set up a four view panel layout and set up the panels to include a Hypershade, Perspective view, Render View and a Top view.

2 IPR setup

- From your Render view panel, select **IPR → IPR Render Globals**.

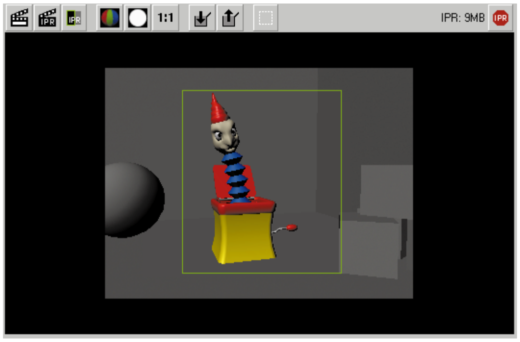
- From the **Anti-aliasing quality** section, set **Presets** to **Production quality**.

  For IPR, you can use the best settings if desired. Your initial IPR rendering will be slower but the interactive updates will still be fast.
- Close the Render globals.

3 Create an IPR rendering

- From your Render view panel, select **IPR → IPR Render → persp**.

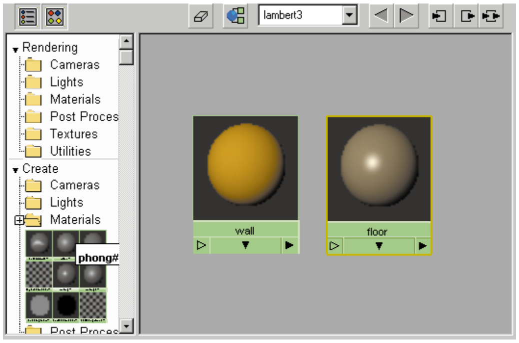
  Now what seems to be a regular rendering of the scene appears. You will learn how to work with this rendering interactively.
- Click drag to select an area of the IPR rendering that includes both walls and the floor. This is the area that will be updated as you make changes.



Initial IPR rendering

4 Create two new materials

- In your Hypershade panel, create a new **Lambert** material node by dragging from the **Create → Materials** section of the visor to the working area.
- Name it *wall* and change its color.
- Create a new **Phong** material.
- Name it *floor* and change its color.



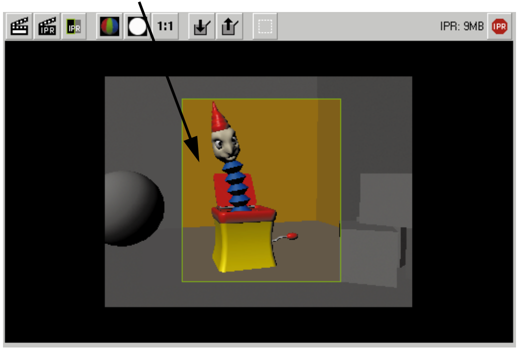
New material nodes

5 Assign the materials

- With your middle mouse button, click-drag the floor material onto the floor surface in the Render View panel within the selected area.

  In the Render View, the new color is updated quickly.
- Repeat for the *wall* material. Be sure to assign it to both wall surfaces.

Drag material into bounded area



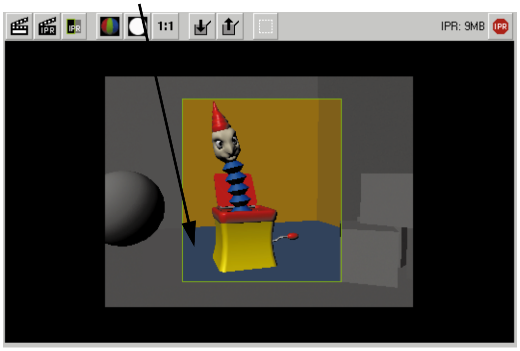
IPR update

6 Edit the Floor material's color

- Open the Attribute editor. Click on the floor in the IPR window. This loads the floor's material into the Attribute editor.
- Click on the Color swatch and in the Color wheel, start trying out different colors.

  As you click-drag around the color wheel, the IPR window will update automatically.

Click to load shader in Attribute editor



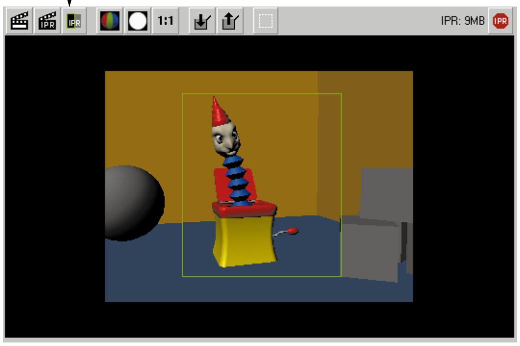
Updated color

7 Refresh the IPR image

- Click on the **Refresh the IPR image** button.

  Now the color and material changes are visible on the whole scene.

Refresh IPR image



IPR update

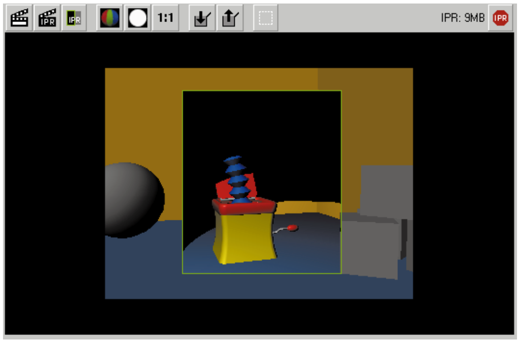
Lighting

You can also use the IPR window to explore different lighting possibilities. You will now set up three light sources with shadows to make the scene more interesting.

1 Change the existing light type

- From the **Rendering** → **Lights** section of the Hypershade visor, double-click on the Directional light icon.
- In the Attribute editor, change the **Type** to **Spotlight**.

  In the IPR update area, you will see that the light is now restricted to a cone angle. This is one of the key properties of a spotlight.



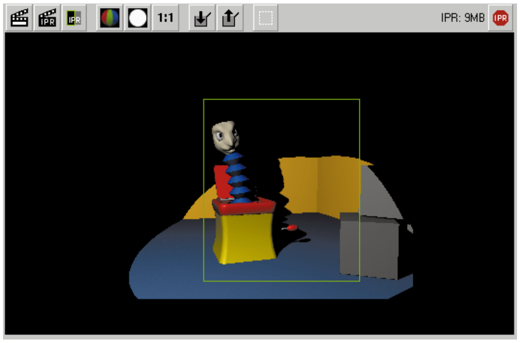
Spotlight in IPR window

2 Edit the light properties

- In the Attribute editor, change the **Cone Angle** of the light to **60**.
- Open the Shadows section and turn **Use Depth Map Shadows** to **On**.

  You will notice that the cone angle adjustment did update in the IPR window but the shadows did not. Some attribute changes may require that you redo the rendering. This takes longer than a refresh.
- Select **IPR** → **Redo Previous IPR Render**.

  Now the shadows are visible.
- Re-select an update area for your IPR view.

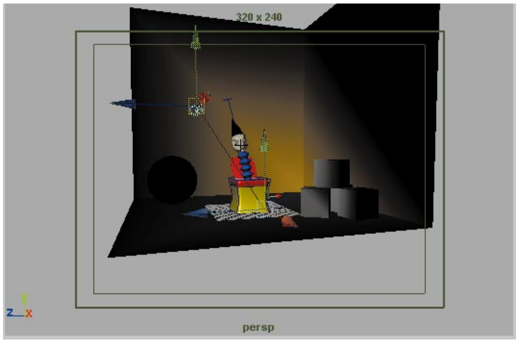


Updated IPR window

3 Edit the Shadow position

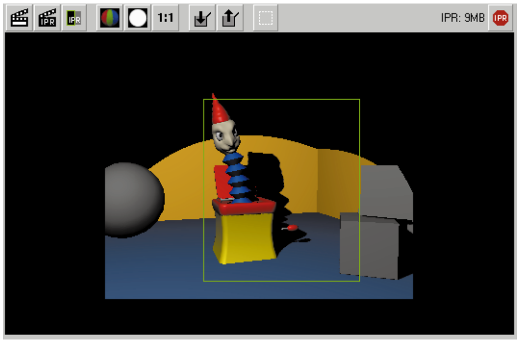
- Zoom out in the perspective view panel.

  You will notice that this does not update in the IPR update area. View changes are not reflected until the rendering is redone. This can be a good thing since you can change views temporarily to edit lights without losing your rendering.
- **Select** the spotlight then select the **Show Manipulator** tool.



Perspective view

- Move the light around until you get a lighting view that you like in the IPR window. You may need to click on the **Refresh** button from time to time to get an overall view of the changes.



Perspective view

- When you are finished, be sure to click in the perspective panel with your middle mouse button then press the **[** key to undo your view changes to get back to your chosen render view.

4 Edit the Shadow properties

- View the spotlight in the Attribute editor and set the following values:

  In the Spot Light Attributes section:

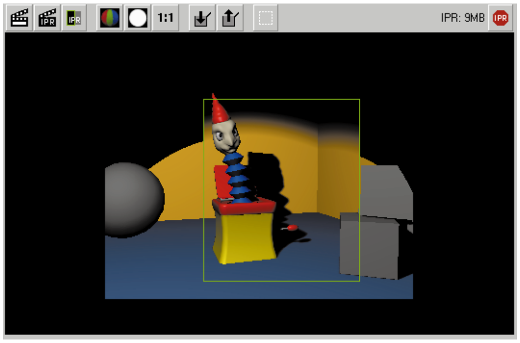
  Penumbra Angle to **5**.

  In the Depth Map shadow attribute section:

  Dmap Filter Size to **3**.

  These settings soften the edge of the spotlight and the shadow.

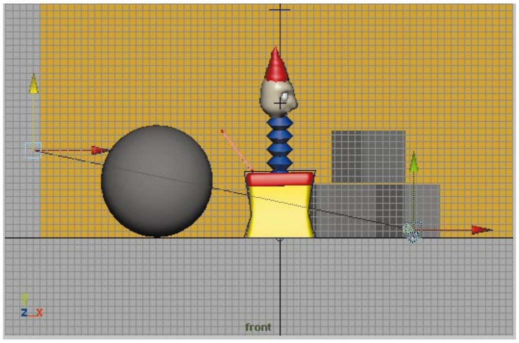
Note: Be careful of setting the Dmap Filter Size too high. It slows down rendering time significantly.



Updated IPR window

5 Add a second spot light

- Select **Lights** → **Create Spotlight**.
- With the **Show Manipulator**, set up the light to look up at Jack from the ground.



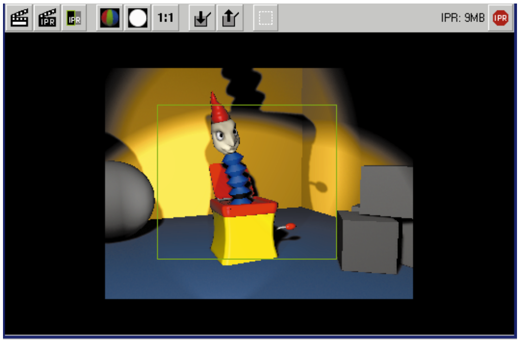
Front view

6 Edit the light

The shadows from this low light will look dramatic on the back wall.

- Turn on **Depth Map shadows** for this light.
- Redo your IPR rendering.
- Set the light's **Intensity** to about **0.75** then play with its light attributes to get the look that you

want for the light. Use the IPR update area to assist you.



Two lights

TEXTURE MAPS

You are now going to add texture maps to some of your materials. Starting with the walls and the floor, you will see how to create color maps that add detail to your rendering.

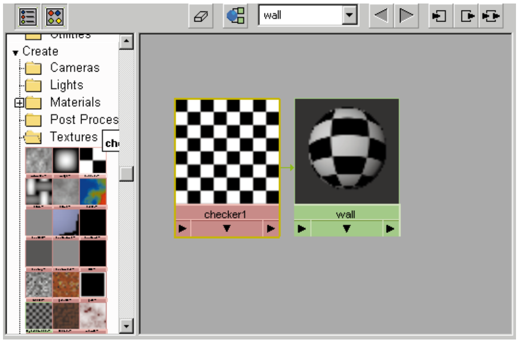
Texturing the wall

You are going to add a simple checker pattern to the wall using one of Maya's procedural textures. A procedural texture gives you a node that has attributes for you to adjust. You can also edit the positioning of the texture to get a more interesting pattern.

1 Add a texture to the wall

- In the Hypershade, select **Graph** → **Clear View** to remove everything from the view.
- From the **Rendering** → **Materials** section of the visor, drag the *wall* shader to the work area.
- From the **Create** → **Textures** section of the Hypershade's visor, use your middle mouse button to drag a checker texture onto the *wall* material.
- When you release the mouse, select **color** from the pop-up menu.

  The texture is now added to the material node in the Hypershade panel and you can see it in the update area of the IPR view.

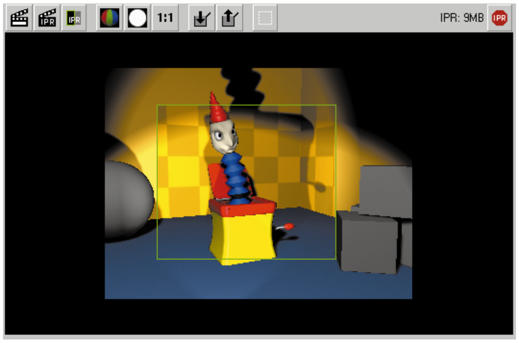


Textured material node

2 Update the textures attributes

The checker texture is a procedural texture that has editable attributes. You can change the color of the opposing parts of the checker.

- Double click on the checker icon in the Hypershade to open up the Attribute editor.
- Click on the **Color1** swatch and change the color as you see fit.
- Repeat for **Color2**.

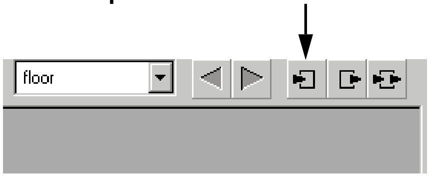


Updated colors

3 Reposition the texture

- In the Hypershade panel, click on the *checker* node to select it then click on the **Show Upstream Connections** button. This displays the texture's placement icon.

Show Upstream Connections



Textured material node

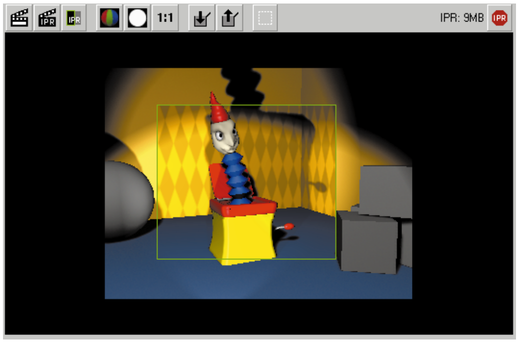
- Double click on the *place2Dtexture* icon in the Hypershade to open up the Attribute editor.
- Set the following:

  Repeat U to **4**;

  Repeat V to **12**;

  Rotate UV to **45**.

  In the IPR update area, you will see the effect that these changes have to the pattern.



Textured material node

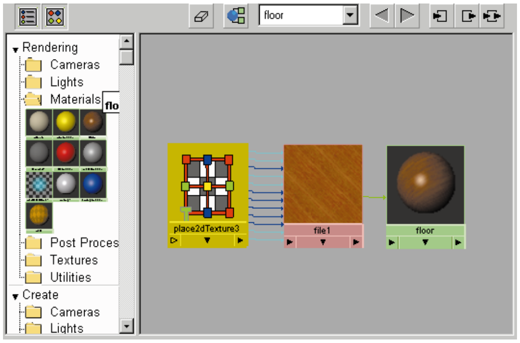
Texturing the floor

To texture map the floor you will use a file texture of a wood pattern. While procedural textures give you a lot of flexibility, a painted or scanned texture can add significant realism to a scene.

1 Create a file texture

- Open the Attribute editor.
- Click on the floor in the IPR update area to place this material in the Attribute editor.
- Click on the **map** button next to **Color**.
- Select **File** from the 2D Texture section of the **Create Render Node** window.
- In the Attribute editor, click on the folder icon next to **Image Name** and choose the *wood.iff* file from the *sourceimages* directory.

  Now you have this texture mapped to the floor.



Textured material node

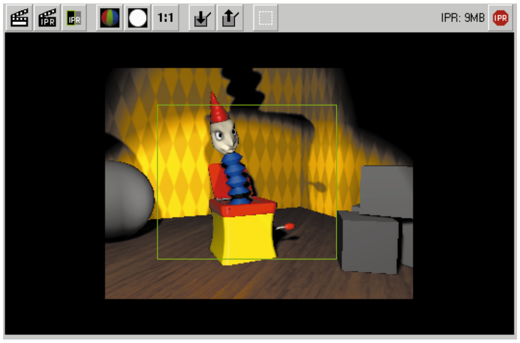
2 Position the texture

- In the Hypershade panel, click on the *checker* node to select it then click on the **Show Upstream Connections** button. This displays the texture's placement icon.
- Click on the *place2Dtexture* icon and in the Attribute editor, set the following:

  Repeat U to **2**;

  Repeat V to **2**;

  Rotate UV to **45**.

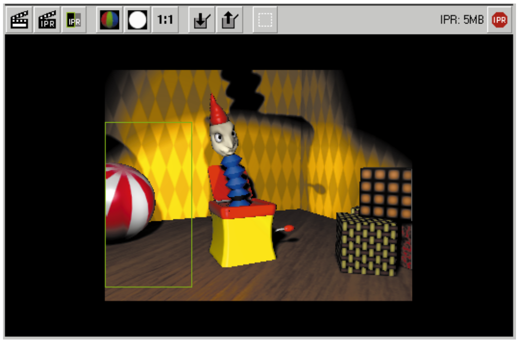


Incorrectly positioned texture

3 Texture the props

- Now use the technique you have just learned to add texture maps to the ball and the blocks. Try out different texture types.

  Remember to move the IPR update area to preview the texture changes.



Correctly positioned texture

TEXTURING THE BASE

You are now going to texture map the base of the jack-in-the-box. You will accomplish this with a projected texture map. Projection maps are positioned using a 3D placement icon instead of sitting directly on the surface. You will set up this map type then explore the implication for the base surface which is a deforming surface.

Projection mapping

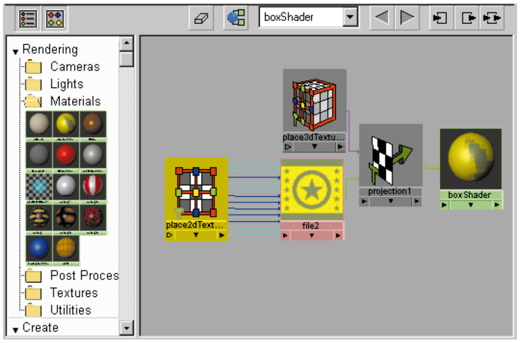
To set up a projection map, you must first set up the material then place the texture in relation to the surface.

1 Set up IPR window

- Go to frame 1.
- Select **IPR** → **Redo Previous IPR Render**. Select an update area.

2 Texture map the base's color attribute

- Drag *boxShader* to workspace.
- Double click to open up the Attribute editor.
- Click on Map button next to color.
- Set **2D Textures** to **As projection**.
- Click on **File**.
- In the Attribute editor, click on the folder icon next to **Image Name** and choose the *box.iff* file from the *sourceimages* directory.



Texture icons

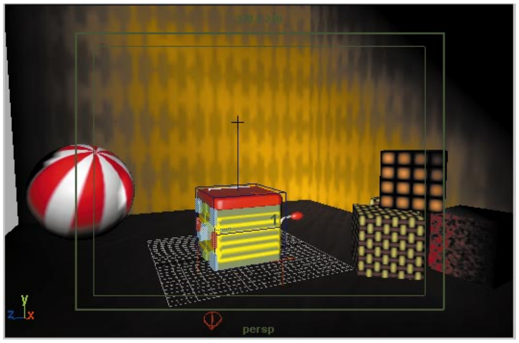
3 Position the texture

- In the Attribute editor, make sure that you are looking at the *projection* node and set the following:

  Set **Projection Type** to **TriPlanar**.

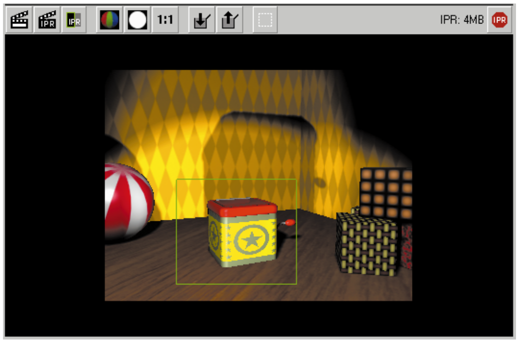
  This map type will project your texture along all three axes.
- Click on the **Interactive placement** button then click on the **Fit to BBox** button.

Now the texture is surrounding the base of the box.



Positioning icons

In the IPR window, you will see that the texture is now mapped to the sides of the box.



Rendered texture

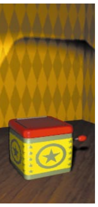
Texture Reference Objects

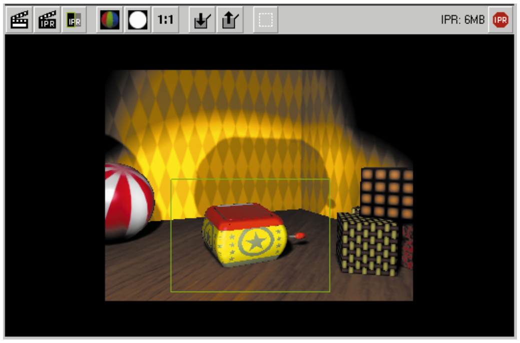
While the projected texture works for frame 1, it will not necessarily work later when the box begins to squash and stretch. You therefore need to set up a non-deforming object as a reference for the texture.

1 Test the texture at frame 55.

- Go to frame 55.
- Redo the IPR rendering.

  You can see that the texture is not following the surface. The stars are not bulging and the top bar of the texture has disappeared. This is because it is using the texture placement node to determine the projection.





Positioned texture

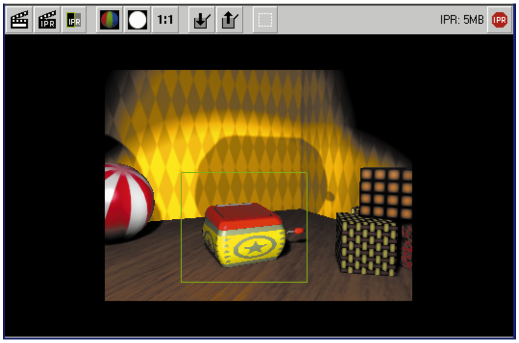
2 Create a texture reference object

- Go to frame 1.
- Select the *base* surface.
- Select **Shading** → **Create Texture Reference Object**.
- Go back to frame 55.
- Redo the IPR rendering.

  Now the texture is working with the surface. What Maya has done is to use a hidden surface to reference the texture. When you deform a surface, there is a hidden undeformed surface that is being referred to during deformations.

  Your projection is now being applied to that hidden object instead of the deforming one.

  You are now ready for the final rendering.



Positioned texture

3 Save your work

- From the **File** menu, select **Save Scene As...**
- Enter the name *jack_final* next to the file's path.
- Press the **Save** button or press the **Enter** key.

Rendering the animation

Once you are happy with your test rendering, it is time to render an animation. This will be accomplished using Maya's *batch renderer*. In preparation, you will add motion blur to your scene to simulate the blur generated in live action film and video work.

1 Set the Image output

To render an animation, you must set up the scene's file extensions to indicate a rendered sequence. You must also set up the Start and End Frames.

- From the Hotbox, select **Window** → **Render Globals...**
- From the **Image File Output** section, set the following:

  File Name Prefix to **jack**;

  This sets the name of the animated sequence.

  Frame/Animation Ext to:

  name.#.ext (for NT)

  name.ext.# (for IRIX);

  This sets Maya up to render a numbered sequence of images.

  Start Frame to **1**;

  End Frame to **120**;

  By Frame to **1**.

  This tells Maya to render every frame from 1 to 120.

2 Turn on motion blur

- In the Render Globals **Motion Blur** section, click on the **Motion Blur** button to turn it on.
- Set the **Motion Blur Type** to be **2D**.

3 Batch render the scene

- Press **F5** to change to the **Render** menu set.
- Use the hotbox to select **Render** → **(Save) Batch Render...**
- Enter the name *jack_anim* next to the path.
- Click on the **Save/Render** button.

4 View the resulting animation

After the rendering is complete, you can preview the results using the *fcheck* utility.

On NT

- Open the fcheck utility by clicking on its icon.
- Select **File** → **Open Animation**.
- Navigate to the `projectTwo\images` folder.
- Select one of the frames of your animation then click **Open**.

On IRIX

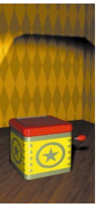
- In a shell window, set your current directory to the `maya/projects/learningMaya/projectTwo/images` directory.
- Type the following:

```
fcheck jack.iff.
```

Conclusion

You are now finished the Jack-in-the-box lesson. Maya's IPR has helped speed up the creative process and allowed you to explore a number of lighting and texturing very quickly.

You are now finished with this project. Next you will build two spaceships and send them into battle.

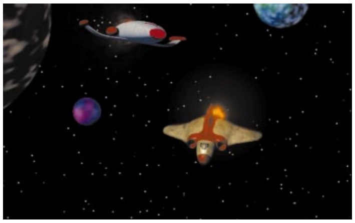


Project Three

Project Three

In this project, you will explore how to model with both NURBS and polgygonal geometry. This will give you a chance to explore some of the possibilities inherent in the two types of geometry. You will then animate the ships along two path curves so that you can easily build and edit a space animation.

You will then create a rendered image of a space scene to use as an animated backdrop. You will then add OptiF/X thrusters, digital photon torpedoes and explosions. Particles will also be used for some effects.



1. A spaceship tries desperately to escape the pursuit of an approaching fighter ship.

2. The fighter shoots one photon torpedo at the fleeing ship.

3. A quick, last second dodge barely clears the resulting explosion.

4. The spaceship breaks away with a trail of smoke emitting from her wing.

12 Polygonal Spaceship

In this lesson, you will build and texture map a polygonal spaceship. Starting with a polygonal cube, you will extrude Poly Faces until you have a completed ship. You will then be able to edit the construction history of the modeling actions to update the model and edit the results.

You will then apply texture projections in order to create UV coordinates on the polymesh. These will let you texture map the ship using a series of texture maps imported as file textures.

Polygon spaceship

In this lesson you will learn the following:

- How to model using polygons
- How to smooth a polygonal mesh
- How to work with procedural modeling attributes
- How to model using lattices
- How to work with procedural modeling attributes
- How to set up files for referencing

Initial set-up

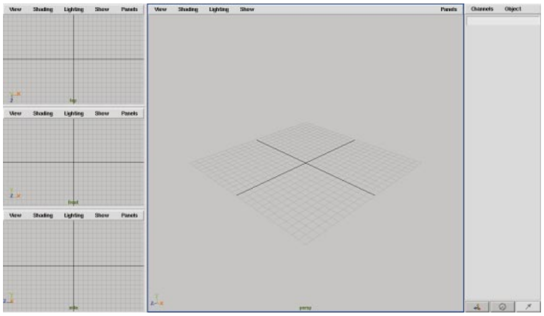
You will use an existing project directory which contains all of the required texture files for the next few lessons.

1 Set the current project

- Select **File → Project → Set...**
- Locate and select the *projectThree* directory in the *learningMaya* directory.
- Click **Set Project**.

2 Set up the modeling panels

- Start a **New Scene**.
- In the view panel menu bar, select **Panels → Saved Layouts → Edit Layouts...**
- In the **Panels** window, open the **Edit Layouts** tab.
- Set the **Configuration** to **4 Pane Left Split**.
- Set the large panel to the Perspective view with the Top, Front and Side view panels along the side.
- Open the **Channel box**.



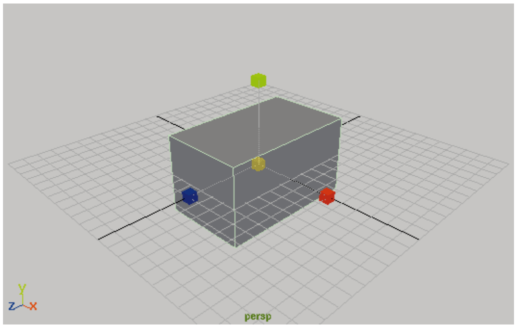
Recommended layout for this tutorial section

Starting the ship

This spaceship will be built starting with a simple polygonal cube. The Poly Faces will then be extruded to create a more complex shape.

1 Place and position a primitive cube

- Select **Create → Polygon Primitives → Cube**.
- Set each view to **Smooth Shade All**.
- Select the **Scale** tool.
- Click-drag on the **Scale** handles to create the following shape:



Scaled primitive polygon cube

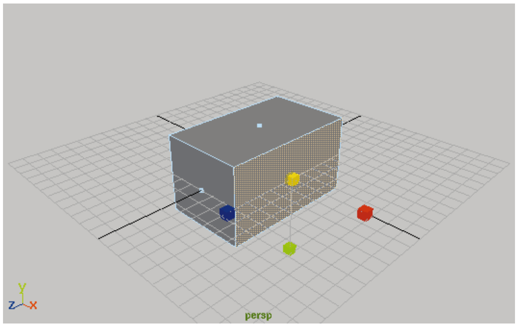
2 Turn on backface culling

During the process of modeling the polygonal spaceship, it is important that you do not accidentally select and modify Poly Faces that are on the opposite side of the object. To prevent this, *backface culling* can be turned on so that the Poly Faces that are facing away from the view are not displayed and cannot be selected.

- Select **Display → Custom Polygon Display - ❐**.
- In the option window, **Set Backface Culling** to **On**.

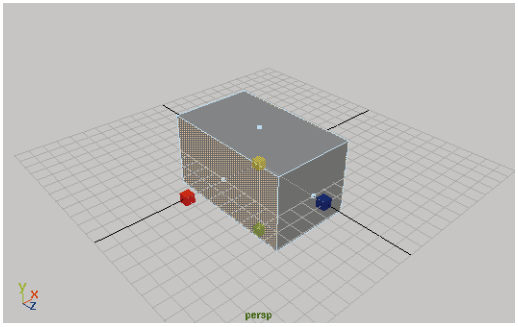
3 Select the two side faces of cube

- Press **F8** to go into **Select by Component** mode.
- Turn off the **Points** selection mask.
- Turn on the **Faces** selection mask.
- **Select** the facet handle on the right side of the cube.



Polygon facet selected

- **Tumble** the view to the other side of the cube.
- Press the **Shift** key and **Select** the opposite facet handle.



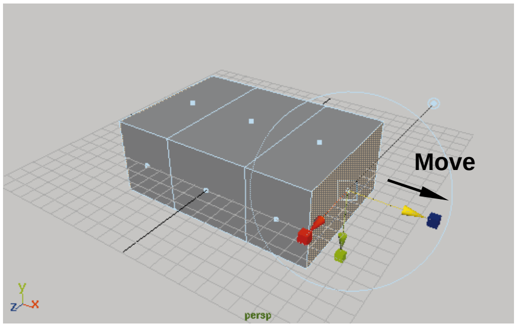
Opposite polygon facet selected

4 Extrude and scale Poly Faces

- Select **Polygons** → **Extrude**.

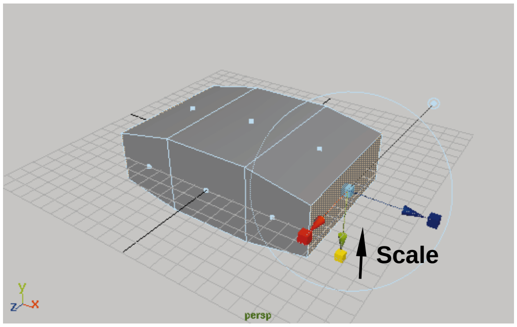
A manipulator appears which gives you access to translation, rotation and scaling all at the same time.

- Tumble the view to get a better view of the manipulator.
- Click-drag on the blue **Move** handle to extrude a new facet.



Opposite Poly Faces extruded

- Click-drag on green **Scale** handle to taper the edges.

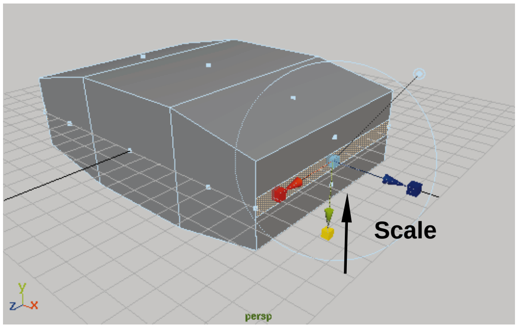


Extruded Poly Faces scaled in one direction

Note: When you manipulate the handle associated with one facet, the other facet reacts equally. Both extrusions are working based on the normals of the original Poly Faces.

5 Add wing base

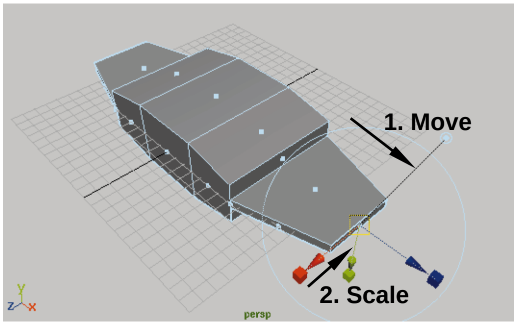
- Press the **g** key to reactivate the **Extrude** tool and start a new extrusion.
- **Click-drag** on the green **Scale** handle to create a long, thin polygon centered about the edge facet.



Thin facet extruded as the base of the wing

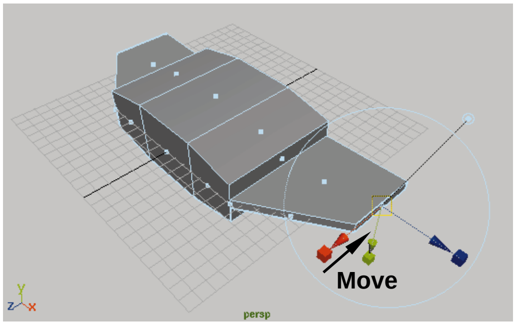
6 Extrude and shape wing

- Press the **g** key.
- Click-drag on the blue **Move** handle.
- Click-drag on red **Scale** handle.



Wing tip extruded and scaled

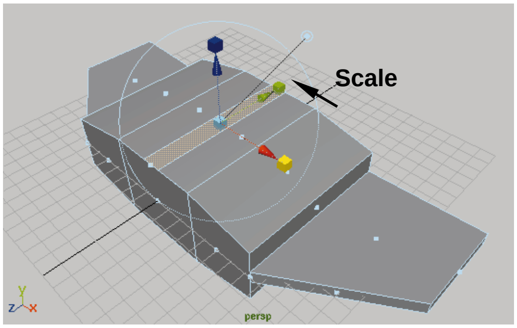
- Click-drag on red **Move** handle.



Wing tip scaled and moved into position

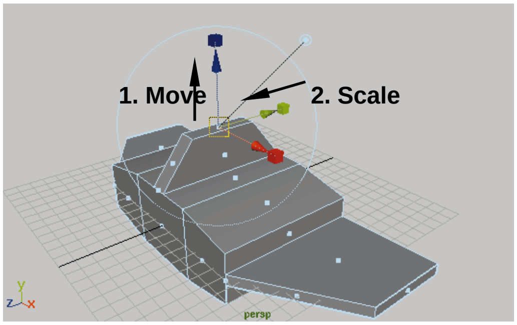
7 Add a top wing

- **Select** the facet handle belonging to the top surface of the shape.
- Press the **g** key.
- Click-drag on red **Scale** handle to create a long thin polygon centered in the top facet.



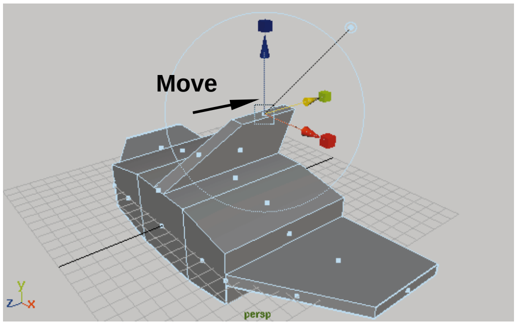
Thin facet extruded as the base of the fin

- Press the **g** key.
- Click-drag on the blue **Move** handle.
- Click-drag on green **Scale** handle.



Fin extruded and scaled

- Click-drag on green **Move** handle.



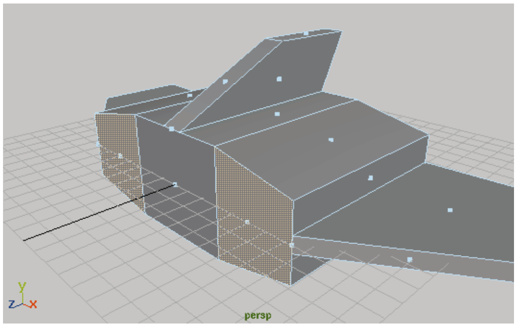
Fin moved back

Air intake ports

In addition to extruding out, you can use the extrusion tool to push a polygonal facet into a surface to create an opening. For the ship's air intake ports, this technique will be used.

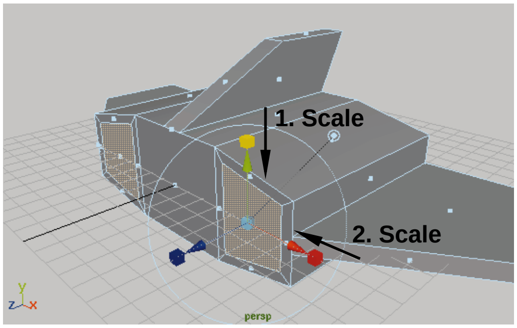
1 Create the front of the ports

- **Select** the facet handle belonging to the side thruster port.
- Press the **Shift** key and **Select** the opposite facet.



Two Poly Faces selected to create thruster ports

- Press the **g** key to re-evoke the **Extrude** tool and start a new extrusion.
- Click-drag on the green **Scale** handle to create a little lip at the edge of the top and bottom of the thruster ports.
- Click-drag on red **Scale** handle to create a similar lip at the side edges of the thruster ports.



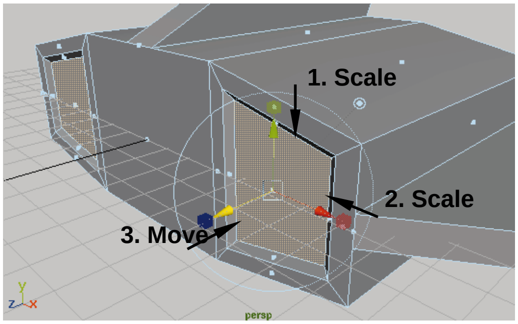
Front edge of the thruster ports

2 Add a lip to the ports

To give the thruster a little bit of an edge, you will next create a little lip using the extrude tool.

- Press the **g** key.

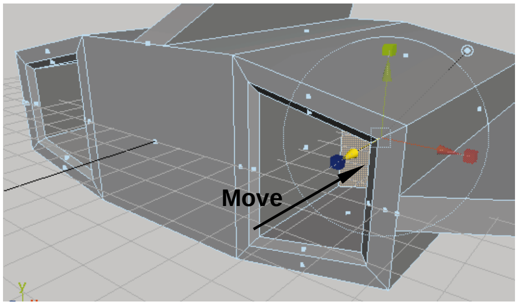
- Click-drag on green **Scale** handle to create a little lip at the top and bottom edges of the thruster ports.
- Click-drag on red **Scale** handle to create a similar lip at the side edges of the thruster ports.
- Click-drag on blue **Move** handle to push this extruded facet back a little.



Lip added to the thruster ports

3 Extrude the hole

- Press the **g** key.
- Click-drag on blue **Move** handle to push this facet back deep into the spaceship.



Hole extruded for thruster port

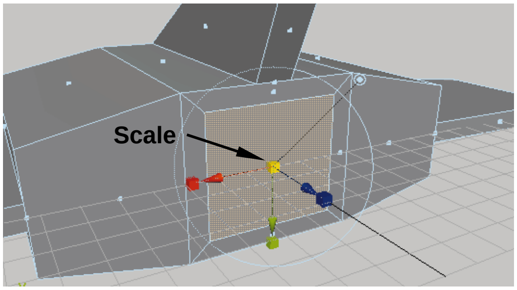
4 Add the rear thruster

- **Tumble** the view around to see the back of the ship.
- **Select** the facet handle belonging to back facet of the ship.
- Press the **g** key.
- Click on one of the **Scale** handles.

The center of manipulator changes to a **Scale** manipulator.

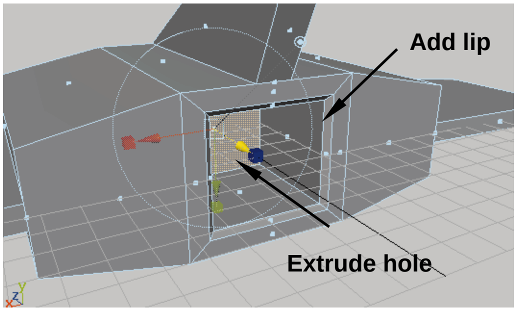
- Click-drag on the center **Scale** handle to create the edge to the exhaust port.

All four sides scale to the same value and proportion.



Front edge of exhaust port

- Repeat the steps used to create the front thruster ports to finish the back exhaust hole.



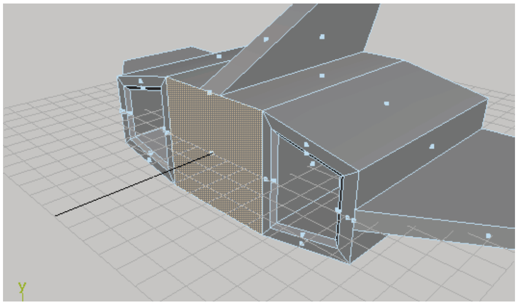
Hole extruded for exhaust port

The front cockpit

You can now continue to use the extrude tool to pull out a cockpit area for the spaceship.

1 Pick the front facet

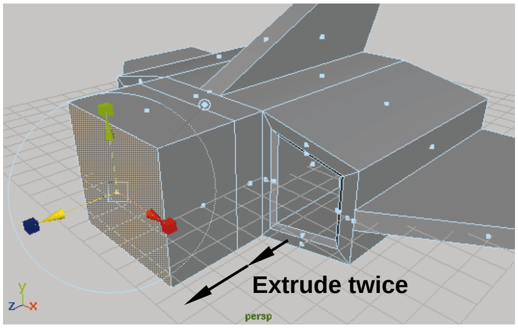
- **Tumble** around to see the front of the ship.
- **Select** the facet handle belonging to front facet of the ship.



Front facet to be extruded for the cockpit

2 Extrude the front facet twice

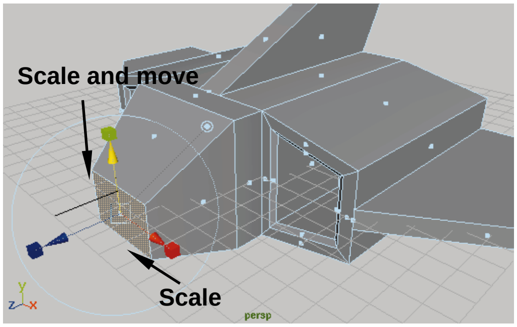
- Press the **g** key.
- Click-drag on blue **Move** handle to pull this facet out.
- Press the **g** key.
- Use blue **Move** handle to pull a second facet out.



Facet extruded twice

3 Scale and rotate the facet

- Use the green **Scale** handle to scale the front facet down.
- Use the red **Scale** handle to scale this facet in.
- Use the green **Move** handle to move this piece down.



Front facet scaled and moved down

Creating a more organic look

At this point, you might be thinking that this will be a very simplistic looking spaceship. You will now apply a smooth node on the polygons to create a more organic look and improve the look of the ship. You will then apply a lattice and deform the lattice points to achieve a more streamlined look.

1 Smooth the spaceship

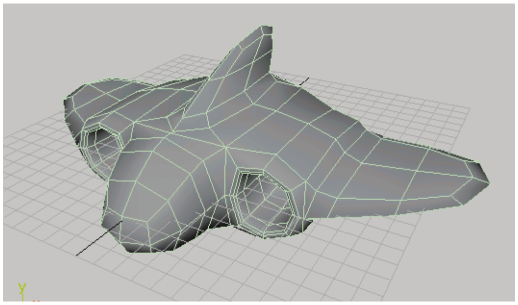
The whole polyset can be smoothed into a more organic shape.

- Press **F8** to go back to **Select by Object Type** mode.

  The facet handles are no longer visible, and the object is now selected.

- Select **Edit Polygons** → **Smooth**.

The planar polymesh that was created out of the series of extrusions is now smoothed and rounded



Smoothed polymesh spaceship

2 Edit the extrude history

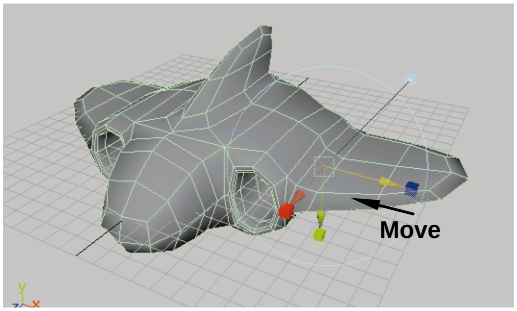
The smooth has been applied as an input node to the polymesh. All of the original extrusions have been recorded in a series of *polyExtrudeFacet* nodes that you can find in the Channel box.

By selecting one of these input nodes and invoking the show manipulator tool, you can edit the history of the extrusion.

- In the Channel box, **Select** the *polyExtrudeFacet1* node.
- Select the **Show Manipulator Tool**.

  The manipulator for the first extrusion is displayed.
- Use the blue **Move** handle to move the extrude in towards the center.

  Both the extrude history and smooth action update, as the exhaust ports become more narrow.



Width of thrusters updated through history

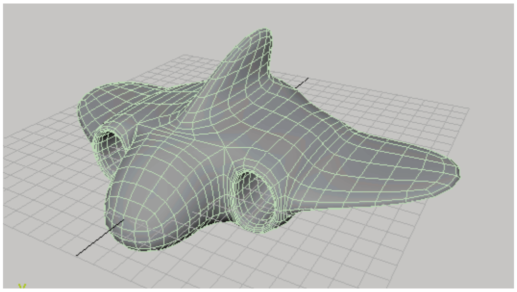
Note: Any of the other extrusions can be adjusted by selecting the relevant input node and using show manipulator, or by editing the values in the Channel box. This is a good example of Maya's procedural modeling capabilities.

3 Edit the smooth history

Since the smooth action is also saved as an input node, you can edit a parameter on this node to increase the smoothness of the ship's surface.

- **Select** the *polySmoothFacet1* input node in the Channel box.
- Set **Divisions** to **2**.

  The detail and smoothness of the spaceship increases.



Level of smoothness increased

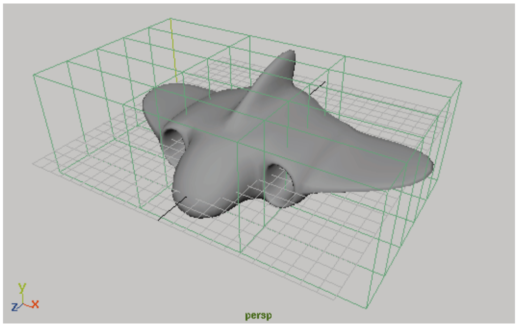
4 Add a lattice

- Change to the **Animation** menu set.
- Select **Deform** → **Create Lattice**.
- In the Channel box, set the following under the *ffd1LatticeShape* node:

  S Divisions to **4**;

  T Divisions to **2**;

  U Divisions to **6**.



Lattice applied to the spaceship

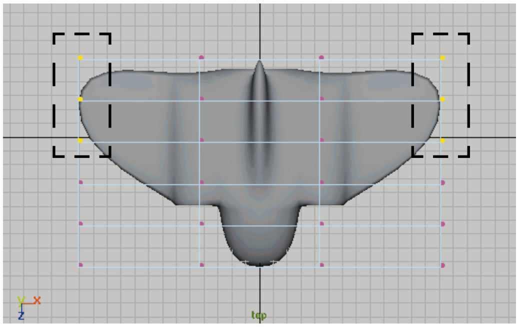
Note: A warning appears in the Command line which indicates that one of the selected items was skipped. This is because you had the *polySmooth1* input node selected. A lattice cannot be applied to this node, so it was applied to the shape node – which is what you want anyway.

5 Edit the lattice points

To reshape the spaceship, you will use the lattice by editing the lattice points. By editing the simple lattice frame, you can easily make edits to the polymesh surface.

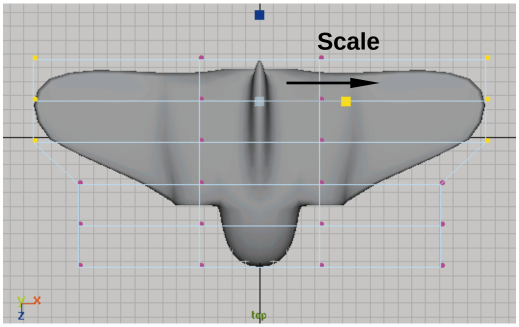
- Press **F8** to go into **Select by Component Type** mode.
- Turn on the **Points** selection mask.

  The **Poly Faces** selection mask should turned off.
- In the Top view, use the **Shift** key and click-drag to **Select** the twelve control points near the tips of the wings as shown.



Selection of lattice points

- Select the **Scale** tool.
- Click-drag on the red **Scale** handle to scale the points out.

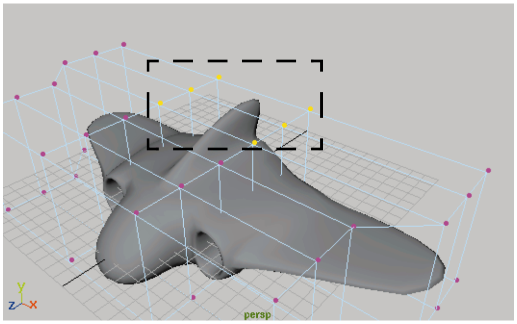


Scaling both wings equally

6 Move the tail

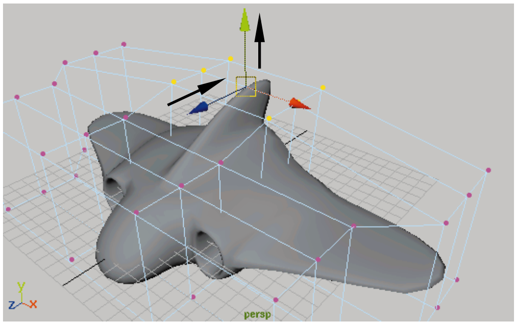
- In the Perspective view, **Select** the six control points closest to the tail.

Make sure that you do not select any of the control points at the bottom of the lattice.



Control points selected

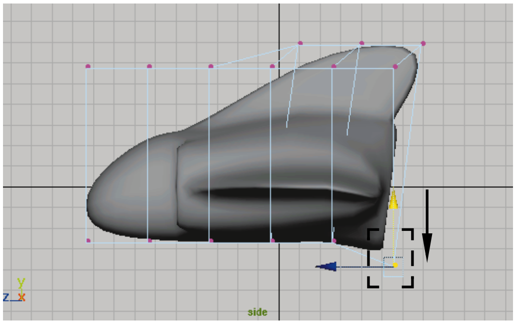
- Select the **Move** tool.
- Use the green and blue handles to **Move** the control points up and towards the back.



Tail area raised

7 Move the lower back edge down

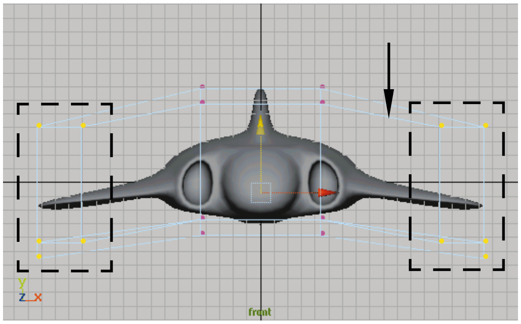
- In the Side view, **Select** the four control points at the back lower edge.
- **Move** these points down.



Back area lowered

8 Move the wings tips down

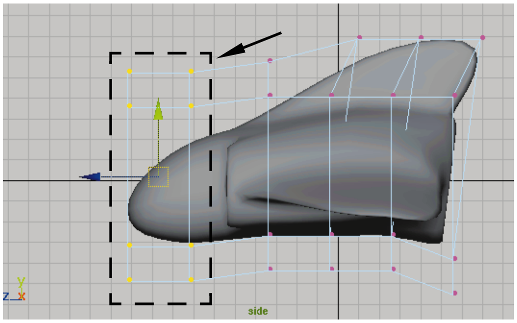
- In the Front view, **Select** all the control points on the right and left sides of the lattice.
- **Move** these points down.



Wing tips moved down

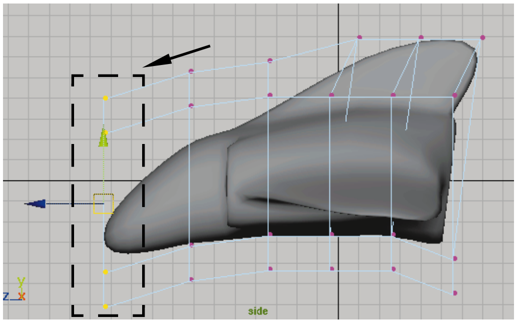
9 Curve the nose forward and down

- In the Side view, **Select** the front two rows of control points.
- **Move** these points forward and down.



Cockpit area moved forward and down

- **Select** the front row of the control points.
- **Move** these points forward and down.



Nose moved forward and down

Finishing the model

Now that you are happy with the shape of the ship, you can delete history. This will remove the input nodes. You could keep them around, but they can add extra complexity to the model which is not required.

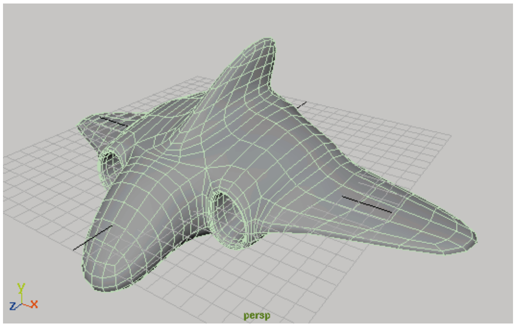
1 Delete history

- Press **F8** to go back to **Select by Object Type** mode.
- **Select** the spaceship.
- Select **Edit** → **Delete by Type** → **History**.

  The lattice and the input nodes are deleted and the history of the polymesh spaceship can no longer be adjusted.

2 Rename the polymesh

- Rename the polymesh *polyShip*.



History deleted on spaceship

3 Save the file

- Select **File** → **Save Scene As...**
- Save the file as *polyShip*.

TEXTURING THE SHIP

You now have a polygonal mesh that requires texturing. Since polygons do not by default have UV parameters, onto which textures can be applied, you can use special polygon tools to assign these kinds of values to the ship.

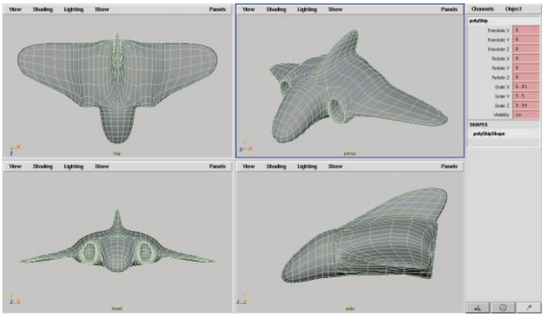
Creating the polyShip shading

The polyShip will be textured using a single shading group that contains several texture maps. The method of positioning the texture on the surface will be accomplished using the polygon texturing tools.

1 Set up view panels

- Select **Panels** → **Saved Layouts** → **Four View**.
- Select **Display** → **Grid** to hide the grid in the view panels.

  This will help you see the texture projection planes.


View Shading Lighting Show Panels
View Shading Lighting Show Panels
Channels Object
View Shading Lighting Show Panels
View Shading Lighting Show Panels

Recommended layout for this tutorial section

2 Create and assign a new shading group

- Open the Hypershade window.
- Create a Blinn Material node.
- **Assign** the Blinn Material to the polymesh object.
- Rename the material node *polyShipBlinn*.
- Rename the *blinn1SG* node to *polyShipSG*.

3 Turn on hardware texturing

If you can use hardware texturing, then it will be a useful tool in this lesson. Otherwise, you will have to test render the model after key steps in the lesson.

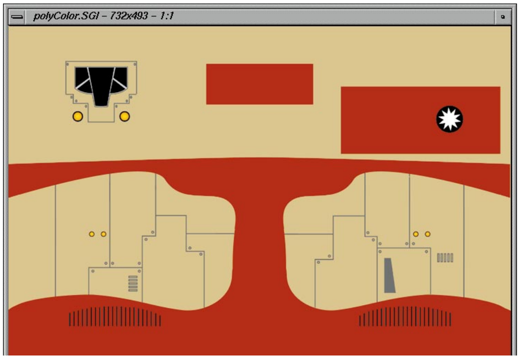
- Turn on hardware texturing in each view panel.

  The display of the model does not change because there are no textures assigned.

4 Map an image file to the color

- Open the attributes for the *polyShipBlinn* node.
- **Map** the **Color** attribute with a 2D **File** texture node.
- Rename the *file1* node *polyColor*.
- Click on the **Folder Icon** button next to **Image Name**.
- In the sourceimages directory of the *projectTwo* project, locate the *polyColor.sgi* image file.
- Click **See Image**.

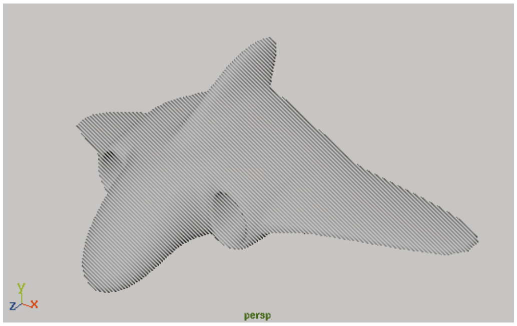
  The image file is a bitmap image that was created in a paint package.



Bitmap file to be used as color texture map

- Close the image and click **Open**.

  If you are using hardware texturing, the polymesh surface changes to a transparent hatching. This is because the polymesh surface does not yet have a UV coordinate system to position the color texture.

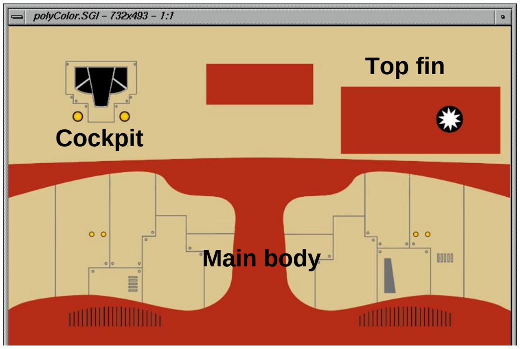


Hardware texturing without UV coordinates

Note: Polygon primitives as well as all NURBS primitives and surfaces have inherent UV coordinates which allow textures to be positioned on a surface. Polymesh surfaces require the UV coordinate system to be projected onto the 3D Poly Faces of the object.

Projecting the texture

The texture you are mapping onto the spaceship has been created as a sort of *decal* sheet. You will be using the polygon projection methods to map the parts of the decal sheet to different parts of the ship. Shown below are the main parts of the ship labeled. Later, you will see how the decal sheet will work.



Ship color texture

1 Turn off backface culling

In order to properly project the texture map onto the polymesh object, it is necessary to be able to select the Poly Faces on opposite sides of the object at the same time. This can be done by turning off backface culling.

- Return to the **Modeling** menu set.
- Select the *polyShip*.
- Select **Display** → **Custom Polygon Display** - ❐.
- Set **Backface Culling** to **Off**.

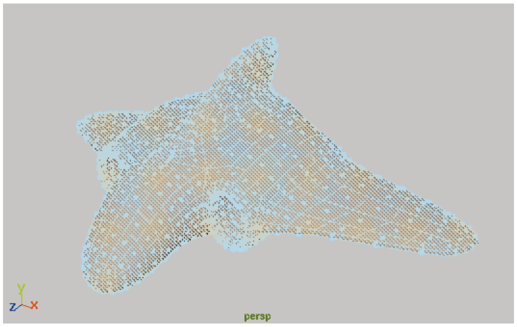
2 Select all the Poly Faces

The Poly Faces of the polymesh surface must be mapped with a projection to create UV coordinates on the polymesh.

- Select the *polyShip*.
- Click on the ship with the **RMB**, and select **Face** from the pop-up marking menu.
- Click-drag a selection box around the entire object to **Select** all of the poly faces.

PROJECT THREE

It is important that you do not miss any Poly Faces. The color of the Poly Faces turn slightly yellow when selected.

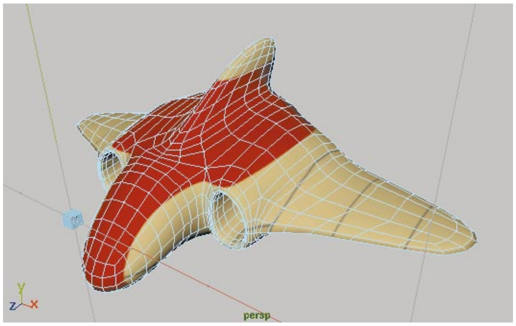


Selection of all the Poly Faces of the polymesh

3 Create planar mapping

- From the main menu bar, make sure that **Edit Polygons → Texture → Assign Shader to Each Projection** is turned off.
- Select **Edit Polygons → Texture → Planar Mapping**.

  A large projection plane icon appears in the in front of the object which projects the texture map from the front of the object. The texture is mapped onto the surface which can be seen with hardware texturing.



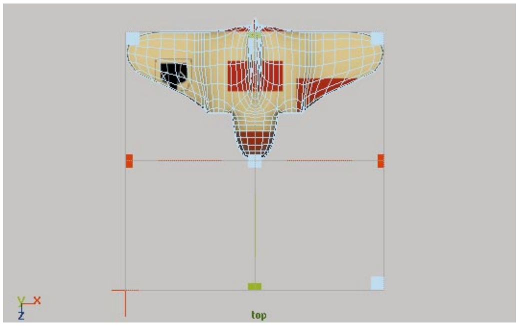
Projection plane manipulator and texture

4 Rotate the direction of planar projection

- Dolly the Top view out.

  The entire texture projection plane should be within view.
- In the Channel box, set **Rotate X** to **-90** and the **Rotation Angle** to **180**.

  The texture is now projected onto the surface from above.



Top view of rotated projection plane

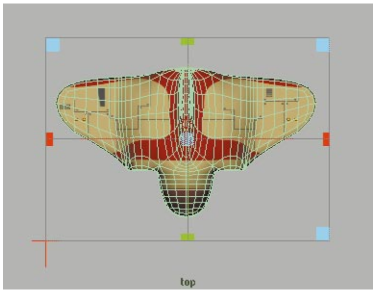
5 Position the planar projection

- Click-drag on the thin green line of the manipulator to move the projection plane along the Z-axis.

The center of the projection plane should be positioned over the approximate center of the polyShip.

- Click-drag on the red and green **Scale** handles to scale the projection plane to slightly larger than the polyShip.

  With hardware texturing, the color texture moves across the surface with the projection plane.



Projection plane moved and scaled

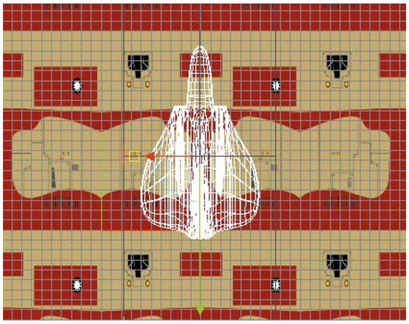
Note: If the projection manipulator disappears, reselect the polyShip, click on the *polyPlanProj1* input node in the Channel box and select the show manipulator tool.

6 Open Texture View

- Select **Edit Polygons** → **Texture** → **Texture View...**

  The window opens with the mapped Poly Faces of the polyShip visible, shown from the view of the texture projection.

- From the Texture View menus, select **View** → **Grid** to hide the grid in the window.



Texture View window

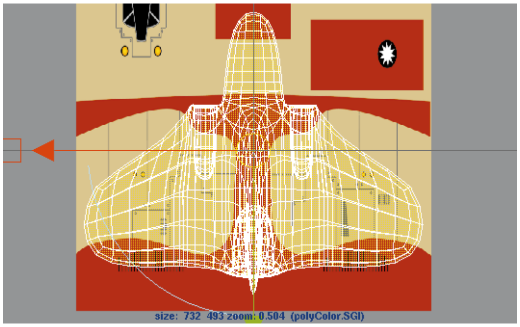
Note: The view of the mapped Poly Faces and the loaded texture are both initially displayed in the Texture View window with a square proportion – regardless of the proportion of the planar projection positioned in the 3D space of the model and the proportion of the texture image file.

7 Position texture icon

You can now move the texture icon to position it properly in relation to the surface. You have the option of using the manipulator in the workspace or the Texture View window.

- Dolly in the Texture View until you see the full extent of the positioning manipulator.
- Use the green box handle to scale the proportion of the polyShip profile.
- Use the green arrow to move the profile into position.

  The polyShip outline should be centered on the texture, with the red box covering the tip of the nose, the wings stretching between the horizontal red bands, and the vertical black lines should overlap the back edge of the wings.

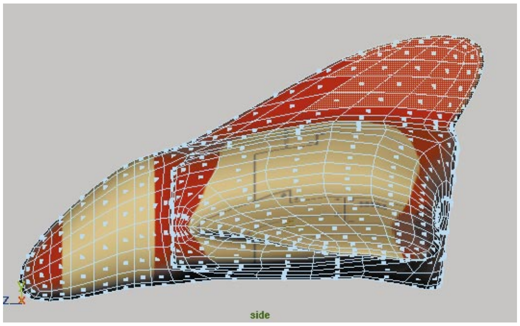


Outline positioned correctly over the texture

8 Examine the texture positioning

- Press the **F8** key to change selection mode.
- Click in one of the view panels to deselect the polyShip.

  You can now see the texture more clearly.
- Tumble in the Perspective view to see the position of the texture on the polyShip surface.



Overall texture positioned on surface

Note: To make any refinements to the texture positioning, select the *polyPlanProject1* input node and select show manipulators.

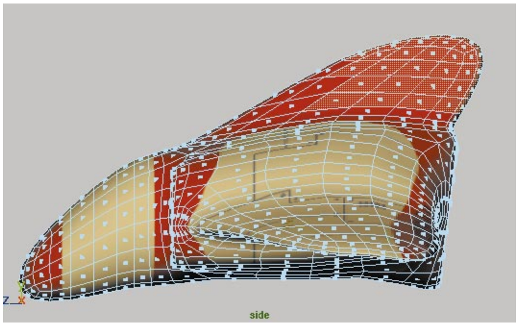
Mapping the fin

The projection of the map down onto the ship does not let you map the fin from the side. You can create a separate projection to these Poly Faces to map the fin with the ship's logo.

1 Select the fin Poly Faces

- Press the **F8** key to return to component selection mode.
- Click on the polyShip to display the Poly Faces.
- In the Side view, click-drag to **Select** the Poly Faces that form the main area of the tail.

  By click-dragging, you select the Poly Faces on both sides of the fin.

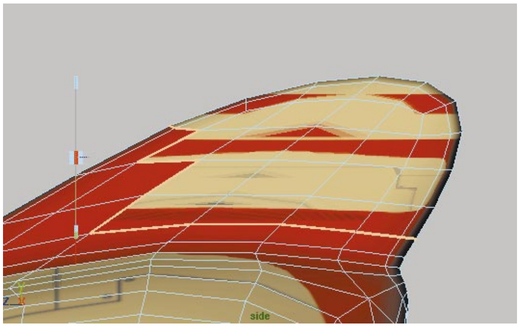


Fin Poly Faces selected for secondary mapping

2 Map the fins with a planar projection

- Select **Edit Polygons** → **Texture** → **Planar Mapping**.

A small projection plane manipulator is positioned in front of the selected Poly Faces, which are outlined with a heavy line. The texture is projected from the front of the polyShip.



Projection plane positioned in front of the Poly Faces

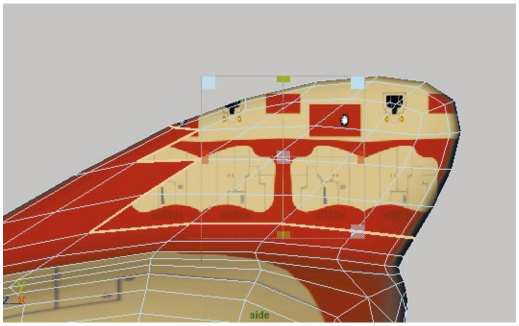
3 Position mapping manipulator

- In the Channel box, set **Rotate Y** to **90**.

  The texture is now projected from the side, and the texture pattern repeats several times across the Poly Faces.

- Click-drag on one of the red horizontal lines of the manipulator to move the projection plane.

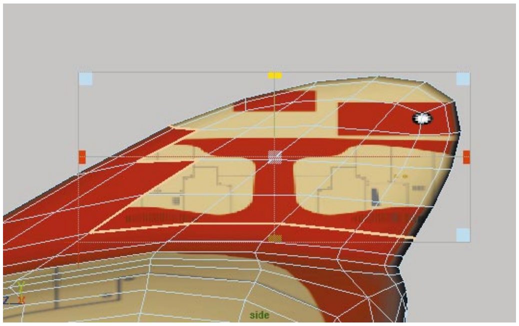
  It should be at the approximate center of the outlined Poly Faces.



Projection plane rotated and centered

- **Scale** the projection plane.

  It should cover the outlined Poly Faces.

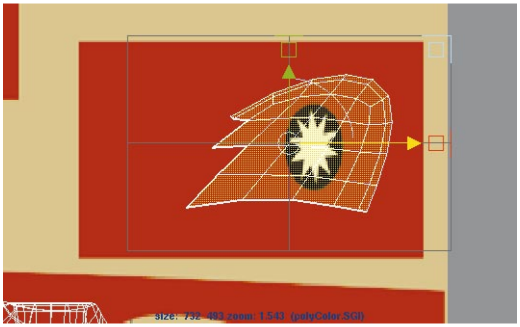


Projection plane scaled to cover facet area

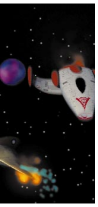
4 Position Texture View

- Dolly the Texture View out so that the entire placement icon is visible.
- Use the blue **Scale** handle to scale the placement icon down.
- Use the **Move** handles to position the icon over the red area in the top right corner of the texture.
- Refine the scale and position of the placement icon.

The Poly Faces should be completely within the red area of the texture and the black and white logo should be centered on the tail profile.

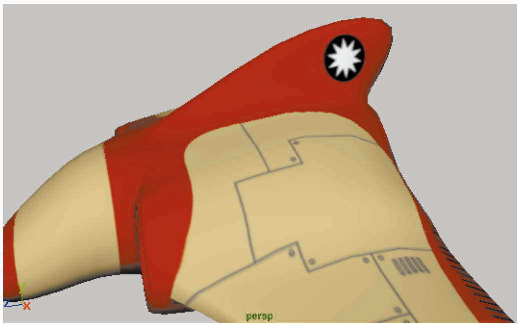


Refined position of fin projection on the texture



5 Evaluate the second projection

- Deselect the polyShip.
- Tumble the Perspective view to see the effect of the second projection.



Second projection on fin

Mapping the cockpit

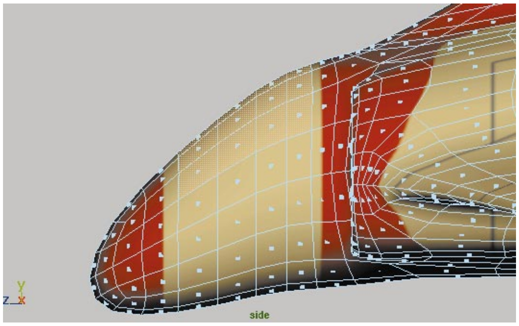
You may be wondering why the cockpit portion of the texture was not part of the original projection. Since the projection goes right through the ship, you didn't want the cockpit appearing on the top and bottom. You will now remap the cockpit onto the top of the ship only.

1 Select nose Poly Faces for cockpit

- Select the polyship.
- Click with **RMB** over the polyship and select **Face**.
- In the Side view, click-drag to **Select** the Poly Faces on the top part of the nose.

  Use the **Shift** key to make multiple selections.

  Make sure that you do not select Poly Faces that are mapped with red areas of the texture.

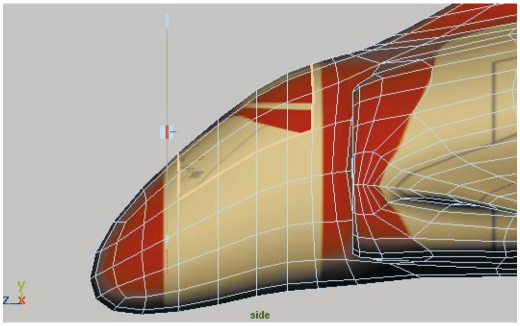


Side view of Poly Faces selected for cockpit

Note: If you do not have hardware texturing, you can see the selected Poly Faces displayed with a yellow hatching in the Texture View window. Make sure that the selected Poly Faces do not overlap with the red bands of the texture.

2 Create planar mapping

- Select **Edit Polygons** → **Texture** → **Planar Mapping**.



Projection plane positioned in front Poly Faces

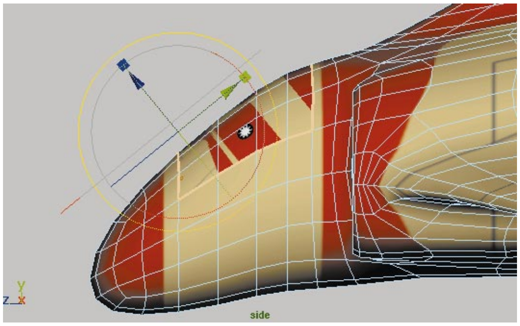
3 Position the projection plane

- Click on the red line handle at the bottom corner of the manipulator.

  The manipulator changes to move, scale and rotate handles.

- Click-drag on the circular blue rotate handle to rotate the projection plane.

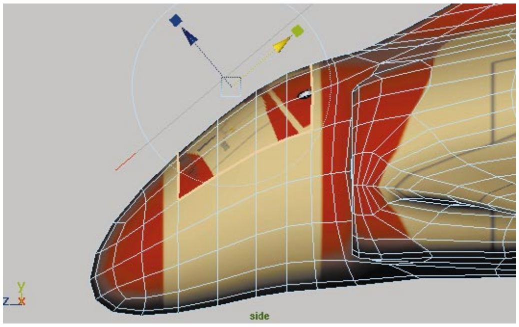
  The plane should be roughly parallel to the top face of the cockpit area.



Rotating the projection plane

- Use the green **Move** handle to move the projection plane up along the nose.

  It should be positioned so that the projection covers the outlined Poly Faces of the cockpit.

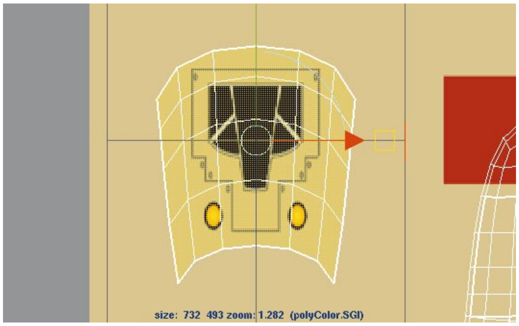


Side view of projection plane in position

4 Position texture view

- In the Texture View window, **Scale** and **Move** the positioning icon, so that it is centered over the cockpit area at the top left corner of the texture.
- Refine the position and scale.

  The outline of the Poly Faces encompasses all the cockpit details of the texture.

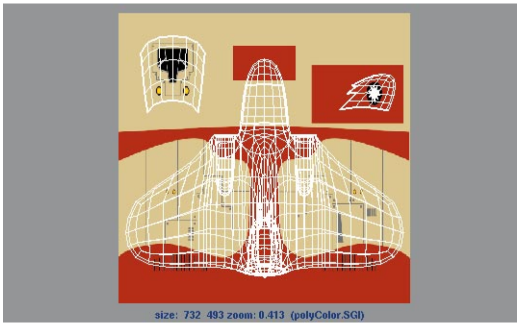


Texture View of cockpit Poly Faces position

5 Examine the three texture projections

You have now used three different planar projections to map one file texture on the Poly Faces of the polymesh of the spaceship. This mapping can be seen in the Texture View window and, if you have hardware texturing, you can examine the results in the Perspective window.

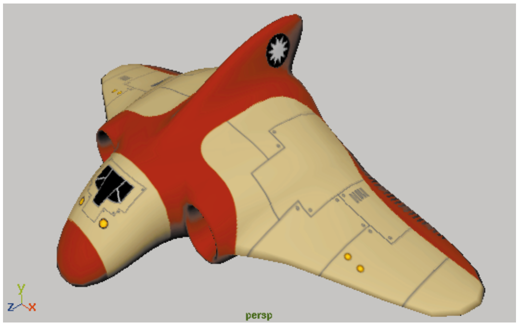
- Press the **F8** key.
- Dolly out of the Texture View window to see the mapping of the three projections on the texture.
- If the projections are not visible, **Select** the polyShip object.



Texture View of final projection mapping

- Deselect the polyShip.
- Tumble the Perspective view to examine the areas where one projection meets another.

  The texture colors should be consistent around the fin and cockpit projection areas.



Perspective view of combined projection mapping

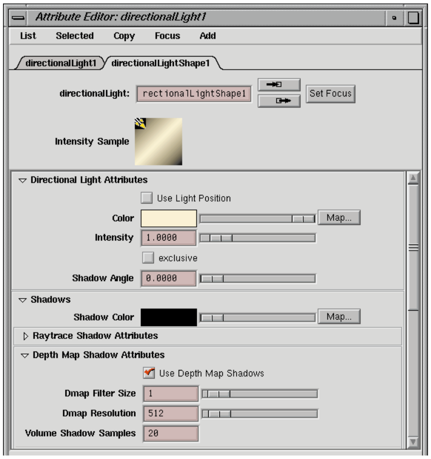
Note: To modify any texture mapping, select the appropriate *polyPlanProj* input node in the Channel box and select the show manipulator tool.

Rendering the ship

To confirm that the texturing is working correctly, you can now render the scene. If you don't have hardware texturing, then this will be your first preview of the results. To render, you will have to add some lights.

1 Add directional lighting

- Change to the **Rendering** menu set.
- Select **Lights** → **Create Directional Light**.
- **Move** and **Rotate** the light so that it is pointing towards the front of the polyShip from above.
- Select **Window** → **Attribute Editor...**
- Set the **Color** to a light yellow.
- Open the **Shadows,** and then the **Depth Map Shadow Attributes** sections.
- Turn on **Use Depth Map Shadows**.



directionalLight: rectionalLightShape1

Intensity Sample

▽ Directional Light Attributes

□ Use Light Position

Color

Intensity 1.0000

□ exclusive

Shadow Angle 0.0000

▽ Shadows

Shadow Color

▷ Raytrace Shadow Attributes

▽ Depth Map Shadow Attributes

☑ Use Depth Map Shadows

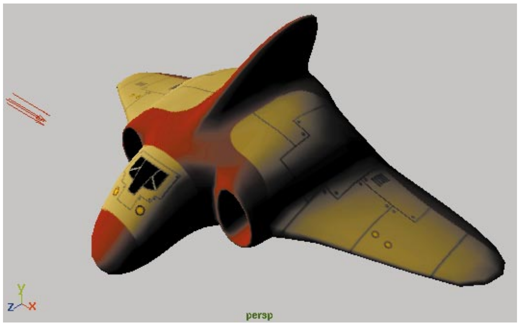
Dmap Filter Size 1

Dmap Resolution 512

Volume Shadow Samples 20

Depth Map Shadows set for directional light

- From the Perspective panel menus, select **Lighting** → **Use All Lights**.



Hardware rendering with directional light

2 Add ambient lighting

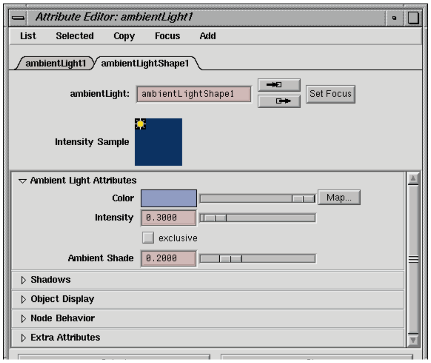
- Create an **Ambient Light**.
- In the Attribute Editor:

  Change the **Color** to blue;

  Reduce the **Intensity** to **0.3**;

  Set **Ambient Shade** to **0.2**.

By reducing the ambient shade, the light will slightly illuminate surfaces that are facing away from the source.



ambientLight: ambientLightShape1

Intensity Sample

▽ Ambient Light Attributes

Color

Intensity 0.3000

□ exclusive

Ambient Shade 0.2000

▷ Shadows

▷ Object Display

▷ Node Behavior

▷ Extra Attributes

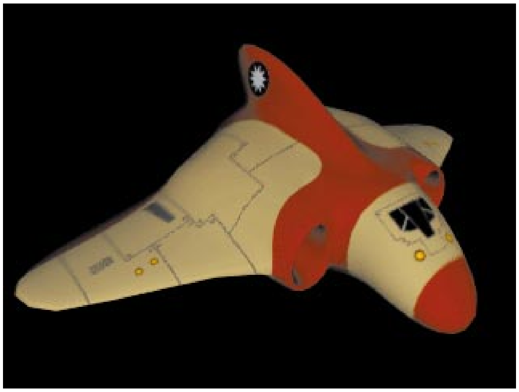
Ambient light attributes

3 Set Render Globals

- Open the **Render Globals** window.
- Open the **Anti-aliasing Quality** section.
- Set the **Presets** to **Intermediate Quality**.

4 Render the scene

- Set up the Perspective view.
- Open the Render View window.
- **Render** the Perspective view.
- **Keep** the image.



Rendering of texture projected onto polyShip

PROJECT THREE

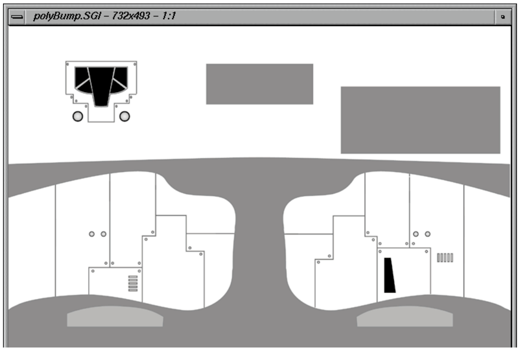
Adding different texture maps

You can now begin refining the shading group to include other types of texture maps. A reflection map for the cockpit, a bump map for the grooved joints, and a specularity map for the different material effects.

1 Map the bump attribute with a file

The next texture map you are going to use is a bump map that raises and lowers parts of the model.

- Open the Attribute Editor for the *polyShipBlinn* material node.
- **Map** the **Bump Mapping** attribute with a **File** 2D texture node.
- Click the *file2* tab.
- Rename the node *polyBump*.
- Click the **Folder Icon** browse button.
- In the *sourceimages* directory, open the *polyBump.sgi* file.



Bitmap file for material bump map texture

Note: Because the color texture map used the default texture placement node settings, you will not need to adjust the values of the placement nodes for the other file textures.

2 Refine the bump map

To create a high quality bump, you should set the depth to define how deep the bump is going to be, and set the filter to define the sharpness or softness of the bump map.

- Open the *bump2d* tab.
- In the **2d Bump Attributes** section:

  Set the **Bump Depth** to **0.25**.
- In the **Effects** section:

  Set **Bump Filter** to **0.2**.

  This setting makes sure that the map is sharp instead of blurred.

3 Map specularity

The next map is going to be used to map the specularity on the surface. Darker areas will produce less specularity while lighter areas produce more.

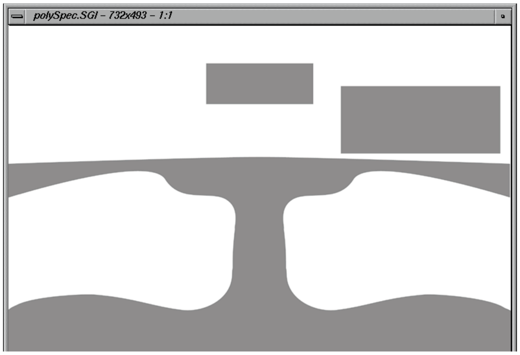
- From the Attribute Editor menus, select **Focus** → **polyShipBlinn**.

  This returns the window to the material node attributes.
- In the **Specular Shading** section set:

  Eccentricity to **0.2**;

  Specular Roll Off to **0.9**.
- **Map** the **Specular Color** attribute with a **File** texture node.
- Rename the *file* node *polySpec*.

- Connect the *polySpec* node to the *polySpec.tif* image file.
- In Render View **Render** → **Redo Previous Rendering**.

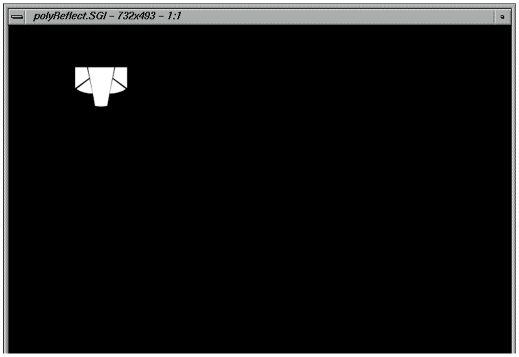


Bitmap file for material specularity texture

4 Map reflectivity

The reflectivity map is quite simple. Where the map is black, you will not get reflections – but where the map is white, you do. This map will not show you any results until you either raytrace the scene or add a reflection map.

- Return to the attributes for the *polyShipBlinn* node.
- Hold down the right mouse button on the **Reflectivity** attribute.
- Select **Create New Texture...** from the pop-up menu.
- Create a **File** texture node.
- Rename the *file* node *polyReflect*.
- Connect the *polyReflect* node to the *polyReflect.tif* image file.
- In Render View select **Render** → **Redo Previous Rendering**.



Bitmap file for material reflectivity texture

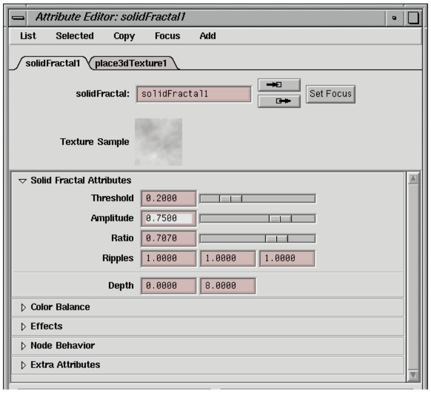
5 Add dirt to the spaceship color

The spaceship textures render to show a clean surface as if the spaceship just left the factory floor. You will now map a 3D fractal map to the color map's Color Gain attribute to roughen up the look of the ship.

- Navigate to the attributes for the *polyColor* file texture node.
- Open the **Color Balance** section.
- **Map** the **Color Gain** attribute with a **Solid Fractal** 3D texture.
- In the **Solid Fractal Attributes** section, set:

  Threshold to **0.2**;

  Amplitude to **0.75**.



solidFractal: solidFractal1

Threshold 0.2000
Amplitude 0.7500
Ratio 0.7070
Ripples 1.0000 1.0000 1.0000
Depth 0.0000 8.0000

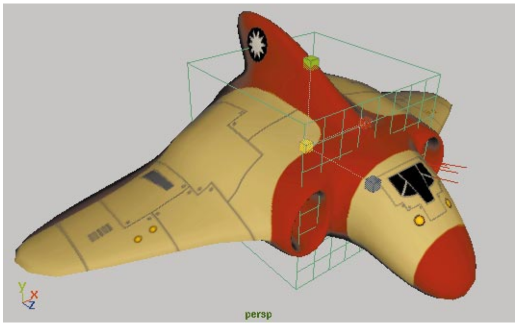
Solid fractal set to add dirt to the color map

6 Scale the 3D texture placement

You need to scale the icon to scale up the effect of the fractal on the surface.

- Click on the *place3dTexture1* tab.
- At the bottom of the Attribute Editor, click on **Select**.

  This selects the 3D placement node in the modeling views.
- In the Perspective view, **Scale** the placement icon to half the size of the polyShip.



Solid fractal 3D texture placement icon scaled

7 Parent the texture placement node

Right now, if you were to animate the spaceship, the dirt would stay behind. You need to parent the node to the ship so that if the ship moves then the texture moves too.

- Press the **Shift** key and **Select** the polyShip.

  It is important that the icon was selected first, ahead of the ship. This means that the icon will be parented to the ship.
- Select **Edit** → **Parent**.

  This parents the placement node into the hierarchy of the polyShip. When the polyShip is animated in a later lesson, the placement node will also be animated. This is necessary so that the fractal dirt does not move on the surface.

8 Render the scene

- In Render View select **Render** → **Redo Previous Rendering**.

  Notice the effect of the bump map, and the difference in the specular qualities between the main color areas of the surface. The layer of grime that is produced by the solid fractal texture is also clearly visible.

Rendering of polyShip with textures and dirt

Note: The effect of the reflectivity texture will not be seen until later in the project, when an environment map of a starfield will be used as the reflected color instead of using raytraced rendered reflections.

9 Save your work

You should save this file as *polyShip*. You will reference this file into another file in a future lesson.

Conclusion

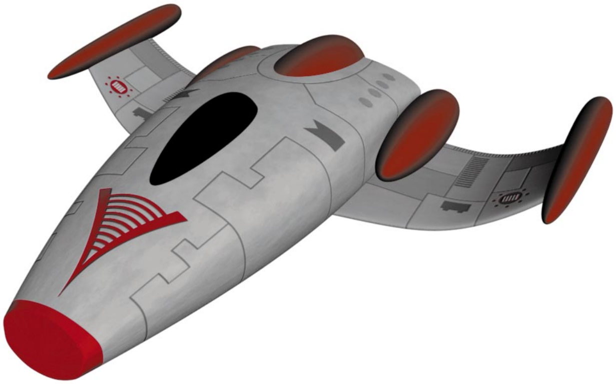
In this lesson, you have learned how to model a complete spaceship out of a primitive cube. In the process, you used polygon modeling tools to create the form. Each tool created an input node that you were able to edit later to change the construction history of the ship.

In the next lesson, you will model another spaceship using NURBS geometry. This will let you compare the two modeling techniques. Later, you will animate the two ships in a space battle.

13 NURBS Spaceship

In this lesson, you will build a second spaceship using NURBS surfaces. You will start by drawing spline curves that will be used to construct the surfaces. Your model will be built out of several pieces that will then be combined to make the final ship.

In the end, you will apply shading groups. For some parts of the spaceship, you will use projection map textures to achieve the desired look.



NURBS spaceship

In this lesson you will learn the following:

- How to draw a spline curve
- How to build a revolved surface
- How to build a birail surface
- How to build a lofted surface
- How to attach surfaces
- How to rebuild a surface
- How to use projection maps

Initial set-up

To create this spaceship, start a new file where you can focus on the new model.

1 Create a new file

- Make sure that your *polyShip* is saved then select **File** → **New**.

2 Set the display options

- Select **Options** → **General Preferences...**
- Click on the **Display** tab.
- In the **Nurbs** section, set the **Shaded Divisions** to **4**.

  Now your surfaces will have a higher surface smoothness and you don't have to press the **3** key every time you create a surface.

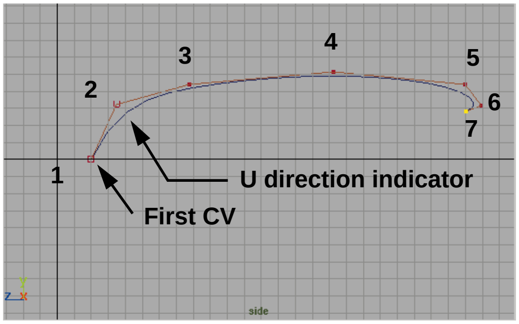
Main thruster

The first part of this spaceship is the main thruster. This will be built by revolving a curve around the Z-axis.

1 Draw the revolve profile

- Select the **Modeling** menu set.
- Select **Create** → **CV Curve Tool**.
- In the Side view, track the view to the left.
- Press and hold the **x** key to use grid snapping and click on the Z-axis line to start drawing a curve.
- Release the **x** key to stop using grid snapping and draw the next six CVs as shown in the following:

  The curve is shown with a box icon representing the first CV and a U icon representing the direction of the curve. Just like X, Y and Z are used to represent spatial dimensions, U is used to represent measurements along a curve.



CV curve points

- Press **Enter** to complete the curve**.**

2 Revolve the curve

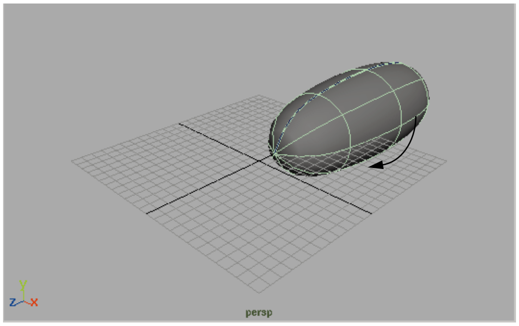
- Select **Surfaces** → **Revolve** - ❐. In the option window, set the following:

  Axis Preset to **Z.**

- Click **Revolve** then **Close**.

  The curve has now been used to generate a revolved surface.

- Use the **5** key to set smooth shading in the Perspective view.



Revolved shape

Construction history

The shape of the surface is dependent on the shape of the curve. You can therefore edit the shape of the surface by editing the curve.

The end of the thruster comes to a point. You will now fix the tangency at the end of the curve to smooth out the revolved shape's end point.

1 Change your Side view display

- In the Side view panel, select **Show → None** then **Show → Curves.**

  Now you can focus on the curve and then preview the shaded surface as it updates in the Perspective view.

2 Get the curve editing tool

- Select **Edit Curves → Curve Editing Tool**.
- Click on the curve in the Side view.

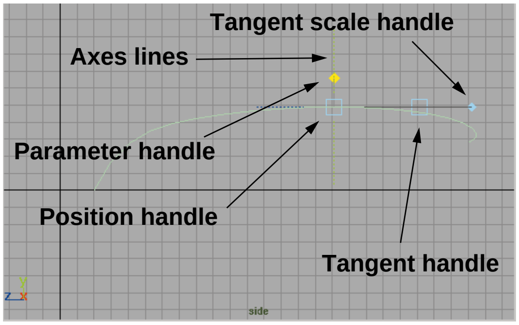
  A manipulator appears with several handles used to edit your curves. These include the following:

  the **parameter handle** which lets you move the editor to a particular point on the U direction of the curve;

  the **position handle** that lets you move the chosen point of the curve and the CVs will update to suit;

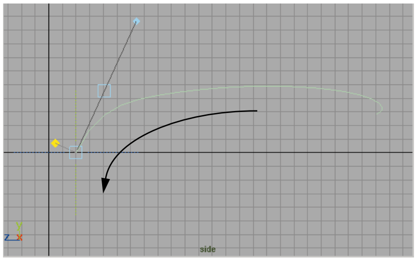
  the **tangent** and **tangent scale handles** let you scale and position the curve's tangent; and

  the **axes lines** can be used to snap the tangent handle to either the X, Y or Z axes.



Curve editing tool

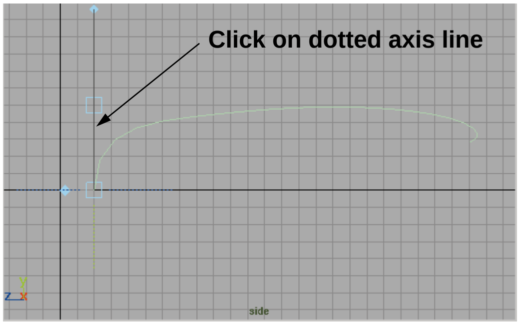
- Click-drag on the yellow parameter handle to move the manipulator to the left end of the curve.



Edited parameter handle

3 Align the end of the curve

- Click on the dotted axis line as shown in the following diagram.



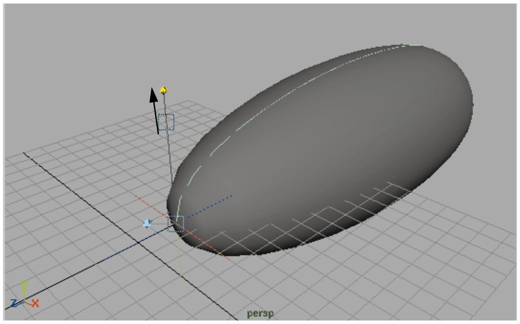
Tangent aligned to axis

The handle's tangent line snaps to align with the axis. In the shaded view, you can see that the end of the shape is more rounded.

4 Scale the length of the tangent handle

- In the Perspective view, click on the tangent scale handle to make it active.

  Click-drag with your middle mouse button to edit the length of the tangent.



Scaled tangent handle

- Press **Enter** to accept the edits.

5 Delete the construction curve

- Select the curve in the Side view.
- Press the **Backspace** key to delete it.

6 Create a new layer and template the main thruster

- Create a new layer called *mainBody.*
- Select the revolved surface.
- Rename it *mainThruster.*
- Assign *mainThruster* to the new layer.
- Template the *mainBody* layer.

7 Save your work

- Select **File** → **Save Scene As...** and name the file *nurbShip*.

THE HULL

The hull of the spaceship will be created by using the birail tool. This tool creates a surface by tracking one or more profile curves along two rail curves.

Drawing character curves

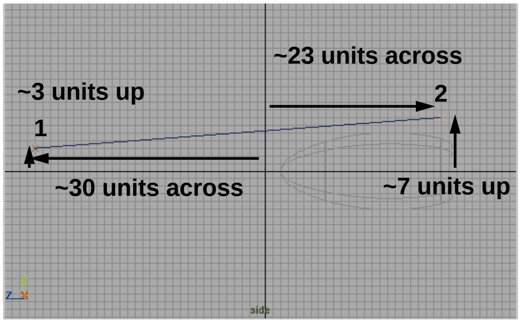
To build the shape of the spaceship's hull, you will start with *character* curves. These curves will define the shape of the surfaces. When working with NURBS models, the quality of your surface is dependent on the character curves.

One approach for creating efficient curves is to start with simple single span curves. When working with NURBS curves, the fewer CVs you have the easier it is to control the curve. You can then insert more spans if you need more complexity.

For now, you will only be building half of the spaceship. Once completed, you will mirror the other half and join together the surfaces.

1 Draw main character curve

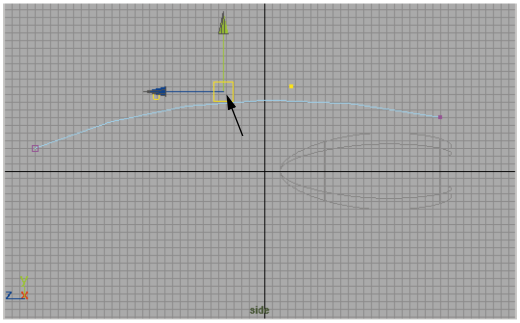
- In the Side view panel, select **Show → All**.
- In the Side view, dolly out to see about **25** units on each side of the main Y-axis line.
- Select **Create → EP Curve Tool**.
- Click and hold in the Side view. Now as you drag, you can use the feedback line to help you place the following points using approximately the values shown below:



Edit point curve

2 Edit the shape of the curve

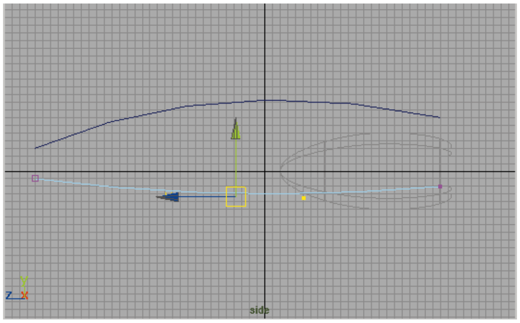
- Press **Enter** to complete the curve.
- Press **F8** to go into component selection mode.
- **Select** the middle two CVs.
- **Move** them up and to the left.
- Press **F8** to go back to object selection mode.



Moved CVs

3 Create the lower rail curve

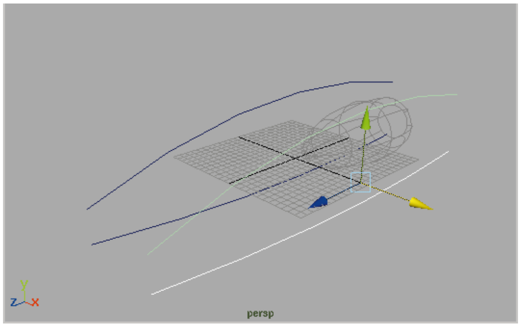
- Use the method outlined in the last two sections to create the following rail curve:



Second rail curve

4 Duplicate and move the curves

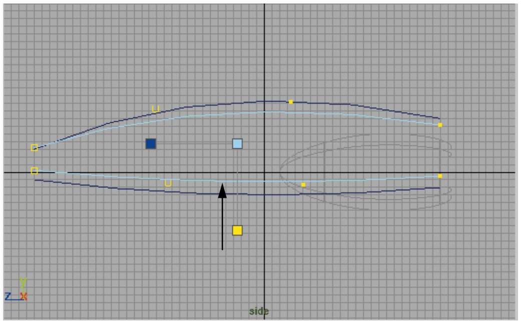
- **Select** the two curves.
- Select **Edit → Duplicate**.
- **Move** the curves about 11 units along the X-axis.



Duplicated curves

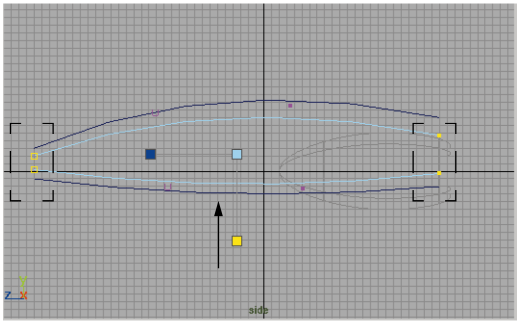
5 Scale the new curve CVs

- Press **F8** to go into component selection mode.
- **Select** all of the CVs on the two curves.
- **Scale** them along the Y-axis to scale them down a little.



Scaled CVs

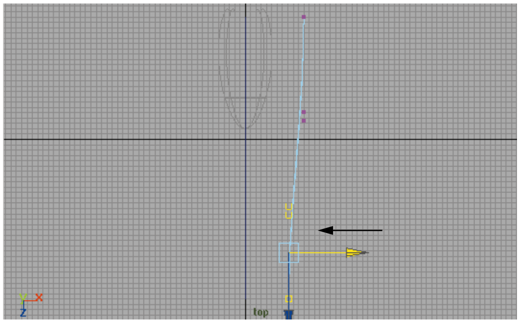
- **Select** the end CVs on the two curves.
- **Scale** them along the Y-axis to scale them down a little.



Scaled end CVs

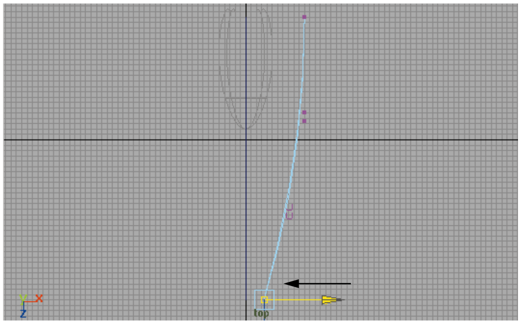
6 Reshape the new curve

- **Select** the first two CVs on the new curves.
- In the Top view, **Move** the CVs about 4 units toward the main axis line.



Moved CVs

- **Select** the first CV on the new curves.
- In the Top view, **Move** the CVs about 2 units toward the main axis line.



Moved end CV

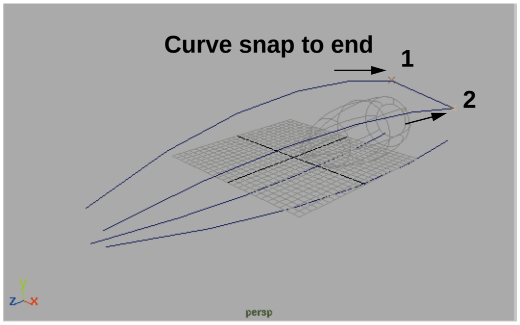
- Press **F8** to go back to Object selection mode.

Profile curves

You now have four rail curves defining half the spaceship body. The next step is to connect the *rails* with *profile* curves. You will be creating three surfaces: a top surface, side surface, and bottom surface.

1 Add the top profile curve

- Select **Edit Curves → EP Curve Tool.**
- In the Perspective view, press and hold the **c** key to temporarily use curve snapping.
- Click and drag onto the back of the center rail curve.
- Drag until this point stops at the end of the curve then release the mouse button.
- Repeat to place a second edit point at the end of the second rail curve.



New edit point curve

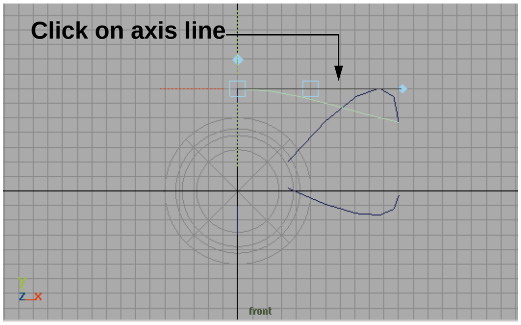
- Press **Enter** to complete the curve.

Tip: By using curve snapping, you are sure that the new curve is touching the two rail curves. This is important to successfully use the birail tool later.

2 Edit curve tangency

To ensure surface tangency at the centre, of the spaceship, you will use the curve editor

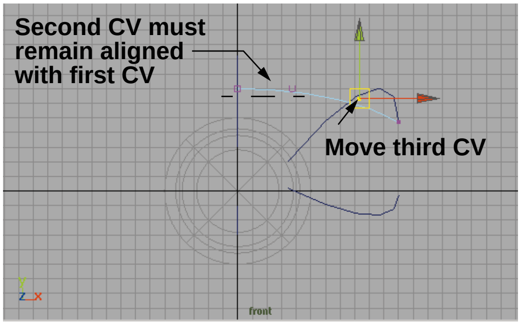
- Select **Edit Curves → Curve Editing Tool.**
- Click-drag on the top handle and drag towards the center axis until it stops at the end of the profile curve.
- Click on the X-axis crosshair (red line) to modify projection.



Aligned curve

- Select the **Move** tool.
- Press **F8** to go into component selection mode.
- **Move** the third CV on the curve to reshape the curve.

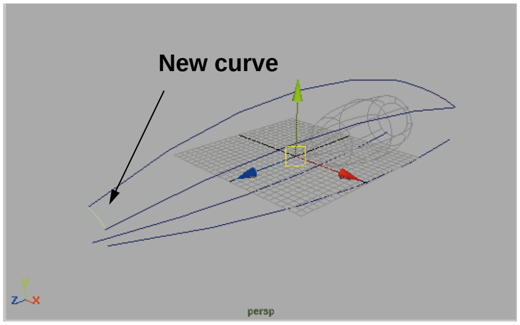
  Don't touch the second CV.



Edited CV position

3 Add the front profile curve

- Use the steps outlined above to add a second profile curve to the front of the rail curves.
- Adjust the curve tangency as outlined above as well.



Second profile curve

Birail surface

A birail surface is created using the two profile curves and the two rail curves. The rails define the sides of the surface while the cross sections are derived from the two profile curves. A birail surface can only be built if all the curves are connected.

1 Create top birail surface

The top surface is now defined by two rail curves and two profile curves. You can now use these to build a birail surface.

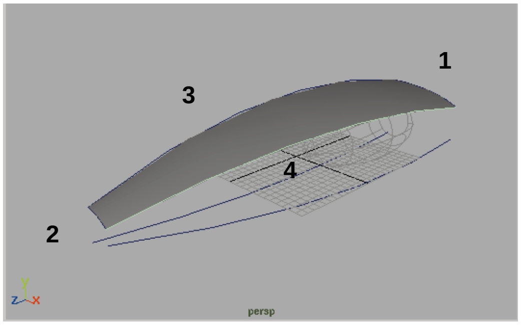
- Select **Surfaces → Birail 2 Tool.**
- Click on the four defining character curves in the following order:

  back profile curve;

  front profile curve;

  center rail curve;

  side rail curve.
- A surface is created along the top of the spaceship.



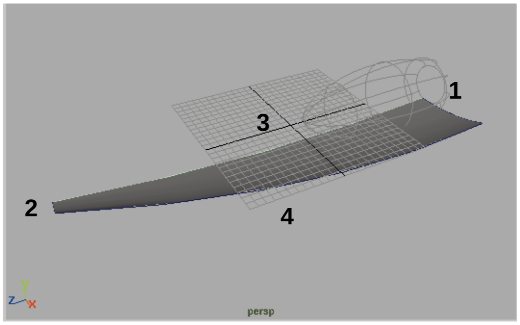
Birail surface showing selection order of curves

Note: If no surface is produced, the profile curves may not be touching the rail curves. Should this be the case, undo your surface creation then go back and rebuild the rails. Also note that you must have **Show** → **Surfaces** on for your view panels to see them correctly.

- If the surface looks correct, press **Enter** to accept the surface.
- **Select** the first surface and the four creation curves.
- Select **Display** → **Hide** → **Hide Selection**.

2 Create bottom birail surface

- Create two profile curves for the bottom of the spaceship using the techniques taught for the first two rail curves.
- Create a birail surface using these curves.



Second birail surface

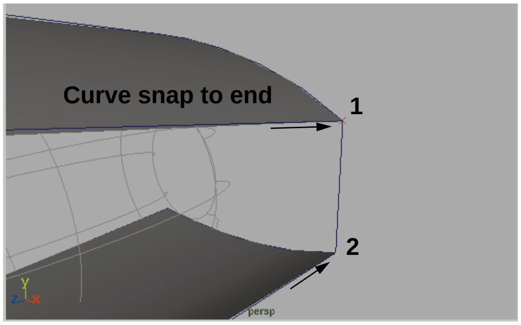
- Select **Display** → **Show** → **Show Last Hidden**.

The side surface

The side of the ship will again be built using a birail surface. This time, you will use three profiles in order to control what the surface looks like at the beginning, middle and end.

1 Create a side profile curve

- In the Perspective view, dolly into the end of the spaceship.
- Select **Curves** → **EP Curve Tool**
- In the Perspective view, press and hold the **c** key to temporarily use curve snapping.
- Click-drag onto the back of the top rail curve.
- Repeat to place a second edit point at the end of the second rail curve.

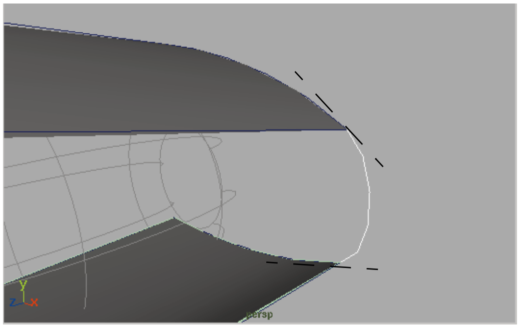


Profile curve

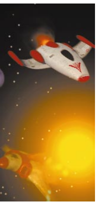
2 Project tangent for profile curve

Before creating the side birail surface, you will create a tangency from the top and bottom surface to the side profile curve.

- **Select** the top surface and the side profile curve.
- Select **Edit Curves → Project Tangent.**
- Repeat these steps using the bottom surface and the profile curve.
- You will notice that the profile curve is now projecting with a tangency from the top and bottom surface.



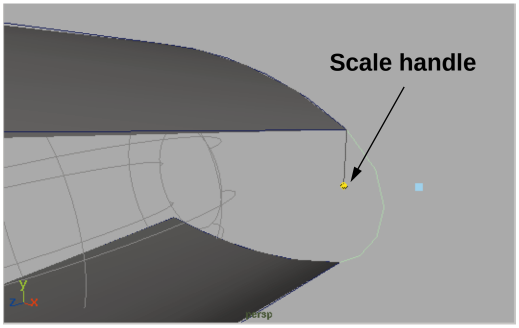
Curve projected on both ends



3 Edit tangency scale

Since the projected tangency curve has history, you can edit the project tangent attributes using the input node and the show manipulator tool.

- **Select** the profile curve on its own.
- In the Channel box, click on *projectTangent1.*
- Select the **Show Manipulator Tool.**
- Click-drag on the **Scale** handle until the **Tangent Scale** is set to about **2.0**.



Scaled tangent handle

As you modify this manipulator handle, you can see the value change in either the Channel box or the Coordinate Feedback line.

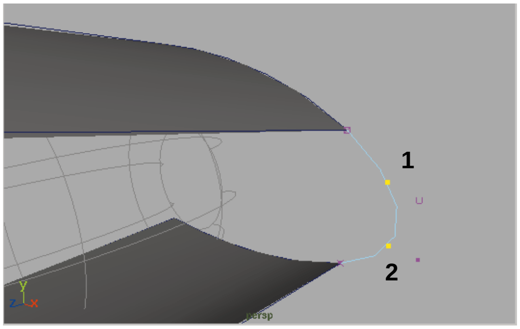
- Use the same workflow to edit the **Tangent Scale** for *projectTangent2* to about **2.0**.

4 Insert CVs to the back side profile curve

You will add two more isoparm spans to this profile curve. This will let you create a more complex profile.

- Press **F8** to go into component selection mode.

- Click with **RMB** over the profile curve and select **Curve Point** from the pop-up menu.
- Click on the profile curve about one third of the curve down to place a curve point.
- Press the **Shift** key and click on the curve to place a second curve point.



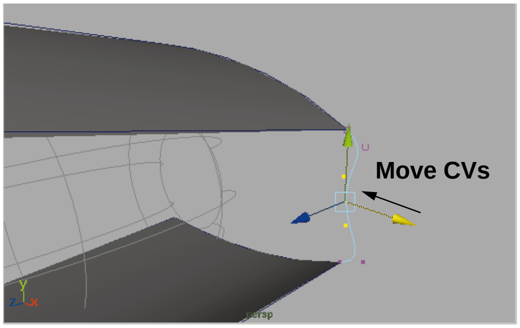
Curve points

5 Insert knots

- Select **Edit Curves** → **Insert knot**.

6 Move the middle curve points

- **Select** the middle two CVs.
- **Move** the CVs in towards the center axis of the ship.

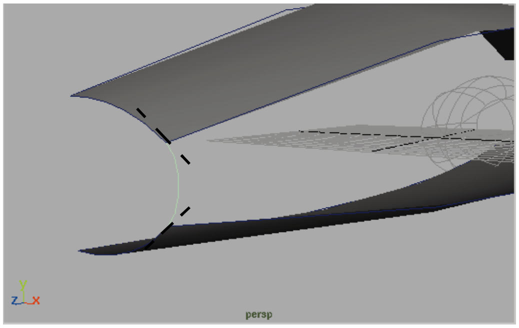


Edited CVs

- Press **F8** to go back into object selection mode.

7 Create the front profile curve

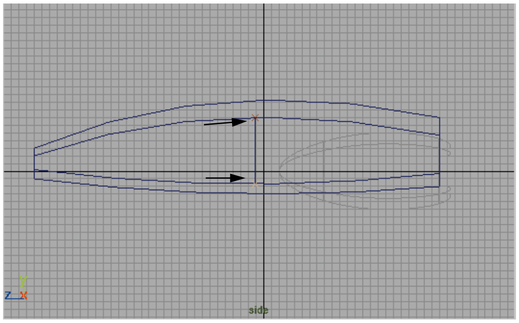
- Draw a single span edit point curve at the front of the ship using curve snapping.
- Use **Project Tangent** to make the curve tangent to the existing surfaces.



Projected edit point curve

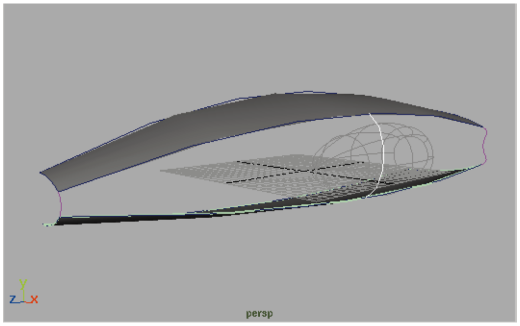
8 Create a middle profile curve

- Draw a single span edit point curve in the middle of the ship using curve snapping.



Curve snap new curve

- Use **Project Tangent** to make the curve tangent to the existing surfaces.

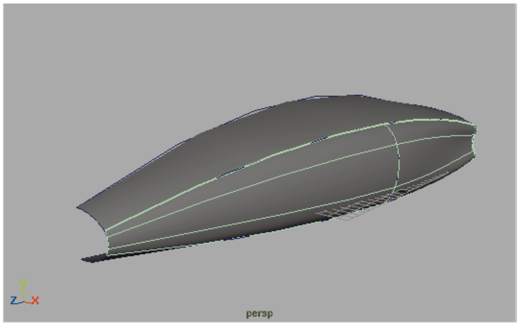


Projected curve

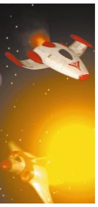
9 Create the side birail surface

- Select **Surfaces** → **Birail 3+ Tool**.
- Click on the three profile curves to select them.
- Press **Enter** to accept these curves.
- Select the outer edges of the existing surfaces as the rail curves.

  The birail surface is created.



Birail surface



Aligning surfaces

When you projected the profile curves off the top and bottom surfaces, you started to build a tangential relationship for the middle surface. Between the profile curves, this relationship is broken. To fix this, you will align and attach the surfaces to make them more continuous.

1 Align and attach surfaces

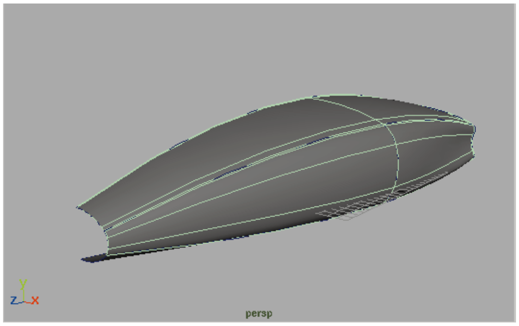
- **Select** the top surface first, and with the **Shift** key select the side surface second.
- Select **Edit Surfaces** → **Align Surfaces -** ❐ and make sure that the following options are set:

  Attach to **On**;

  Modify Position to **Second**;

  Modify Tangent to **Second.**
- Press **Align.**

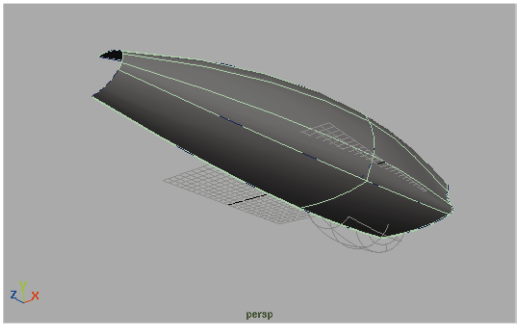
  The top and side surfaces are one surface now.



Aligned and attached surfaces

- **Select** the bottom surface first, and the middle surface second.
- Select **Edit Surfaces** → **Align Surfaces**

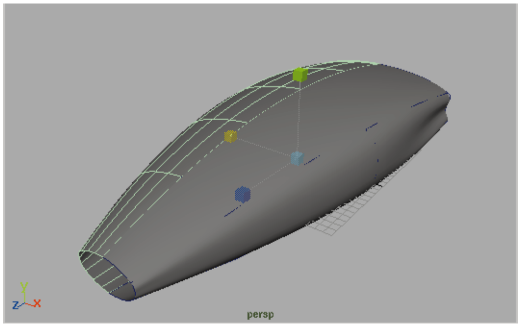
  All three surfaces are now one complete surface.



Aligned and attached surfaces

2 Duplicate and mirror the surface

- **Select** the new surface.
- Press **Ctrl-d** to **Duplicate** it.
- Enter **-1** in the **Scale X** field in the Channel Box.

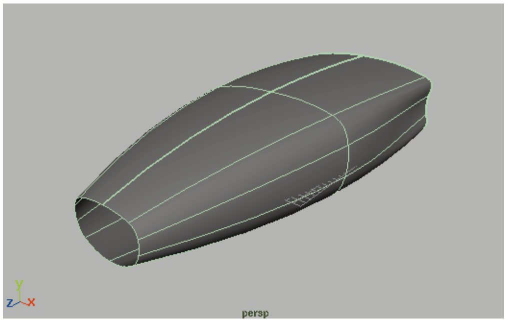


Mirrored surface

3 Align and attach the two halves

- **Save** your work before this next step.
- **Select** the two surfaces.
- Select **Edit Surfaces** → **Align Surfaces.**

You may have noticed that the seam of the two surfaces is on top of the ship. You can control this by selecting the edges that you want to align and attach.



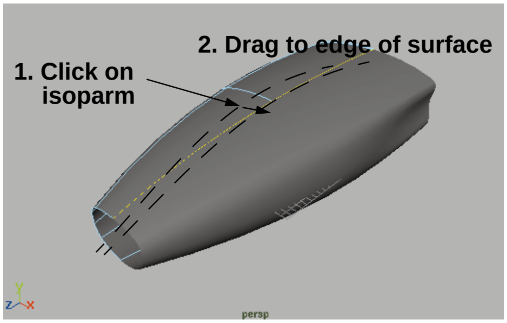
Aligned surfaces with seam on top

4 Undo the last align

- Undo your last step or revert to your last saved.

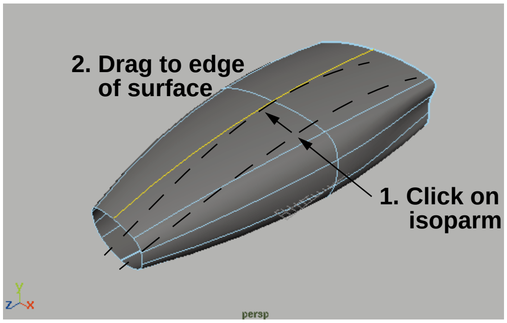
5 Select the edge isoparms

- **Select** the new surface.
- Click on the new surface with your right mouse button and select **Isoparm** from the pop-up menu.
- Click-drag on one of the longitudinal isoparms and drag to the edge of the surface.
- When you release your mouse button, a yellow line will indicate your selection.



Selected isoparm

- Click on the other surface with your right mouse button and select **Isoparm** from the pop-up menu.
- Press the **Shift** key and click-drag on one of the longitudinal isoparms and drag to the edge of the surface.
- When you release your mouse button, a yellow line is placed at the newly selected edge.



Second selected isoparm

6 Align and attach the two halves

Now that you have chosen which edges are to be aligned, you will get the proper results.

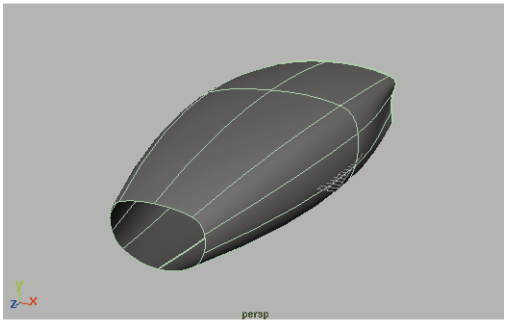
- Select **Edit Surfaces** → **Align Surfaces -** ❐ and make sure that the following options are set:

  Attach to **On**;

  Modify Position to **Both**;

- Press **Align** then **Close.**

  The seam will now appear at the bottom of the ship.



Attached surfaces with seam at bottom of ship

7 Close the surface

You will now close the surface so that the edges are connected properly.

- Select the nurbs ship.
- Select **Edit** → **Surfaces** → **Open/ Close Surfaces.**

8 Delete history

- With the surface selected, select **Edit** → **Delete by Type** → **History**.
- In the Side view, select **Show** → **None** then **Show** → **Curves**.
- In this view, **Select** the various profile curves.
- Press the **Backspace** key to delete the curves.

TRIM SURFACES

The surfaces you have built so far have all been created so that you can add them together. If you want to cut a section out of your surface, then you will need to trim it.

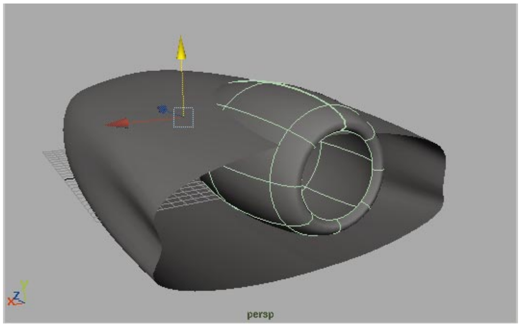
Trimming requires that you define a trim region on the surface using curves on surface. In an earlier lesson, you made the surface live and then drew the curve on surface directly. In this lesson, you will create curves on surface by intersecting existing surfaces.

The back surface

Slightly recessed from the back of the ship's hull, you will next need to place a surface. This surface will intersect the hull and the thruster. You will create curves on surface and then trim.

1 Untemplate and move the thruster

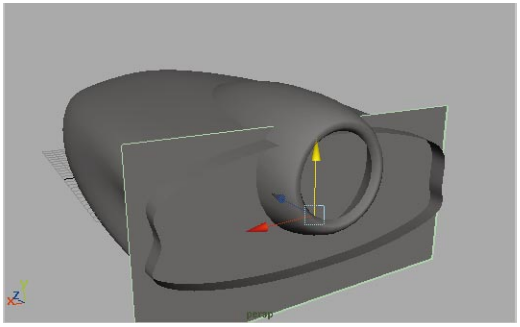
- Untemplate the *thruster* layer.
- Move the thruster until it intersects the top of the ship's hull.



Untemplated thruster

2 Create and place a primitive plane

- Select **Create** → **NURBS Primitives** → **Plane**.
- **Move** the plane along the Z-axis to the back of the ship.
- **Scale** the plane along the X and Z axes to make it about the same size as the ship.
- **Rotate** the plane 90 degrees around the X-axis.
- **Move** it until it is intersecting the end of the hull.
- Select **Edit** → **Delete by Type** → **History**.



New plane intersecting main body

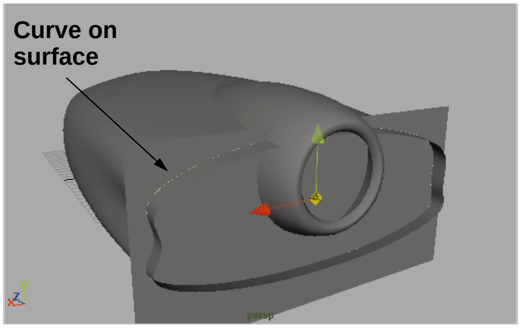
3 Intersect the plane and the body

- Press the **Shift** key and select the ship's hull.
- Select **Edit Surfaces** → **Intersect Surfaces** - ❐ and set the following options:

  Create Curves for to **First Surface;**

  Curve Type to **Curve on Surface**.
- Click **Intersect** then **Close**.

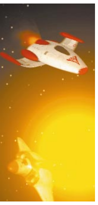
PROJECT THREE

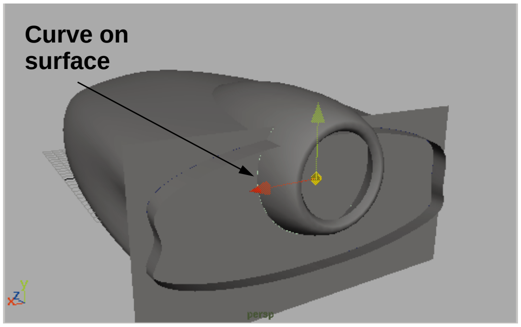


Intersecting curve on surface

4 Intersect the plane and the thruster

- **Select** the plane then press the **Shift** key and **Select** the thruster surface.
- Select **Edit Surfaces → Intersect Surfaces**.



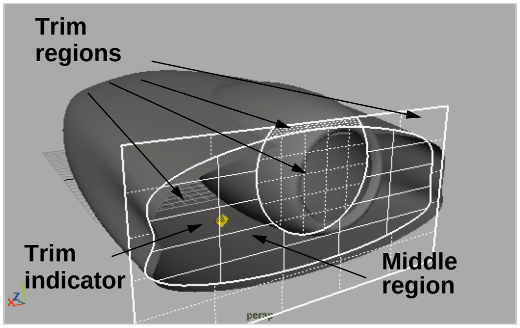


Second curve on surface

5 Trim the plane surface

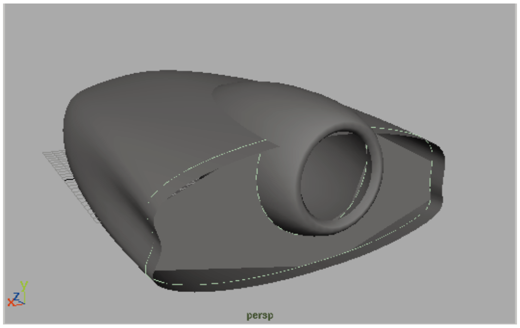
- **Select** the plane.
- Select **Edit Surfaces → Trim Tool.**

  The surface is displayed with its trim regions identified. This lets you choose which areas to keep or discard.
- Click in the middle area to place the trim indicator. This means that this area will be kept.



Trim regions

- Press **Enter**.



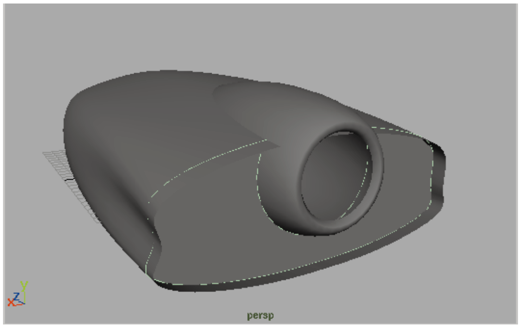
Trimmed surface

6 Increase the surface smoothness

You will now increase the smoothness so that the surface will look better in the interactive view panels.

- Select **Display → NURBS Smoothness → Custom** - ❒ and in the **Shaded** section, set the following:

  Surface Div per Span to **8**.
- Click **Apply** then **Close**.



Increased surface smoothness

7 Increase the surface tessellation

While the smoothness will look good interactively it may not render properly. You must set the surfaces tessellation to ensure that the trimmed surface will render nicely.

- Press **Ctrl-a** to open the Attribute Editor.
- Click on the *trimmedSurfaceShape1* tab in the Editor.
- Open the **Tessellation** section, and set the following:

  Display Render Tessellation to **On**

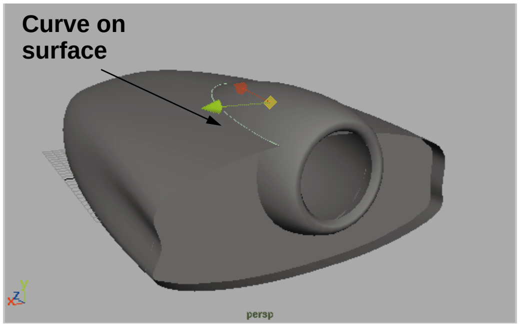
  Smooth Edge to **On**

  You can now see the effect that the smooth edge has on the surface. You do not need to have the **Display Render Tessellation** set on to have it render with the settings. This is only a visual feedback
- Set **Display Render Tessellation** to **Off.**

8 Intersect the body and the thruster

- **Select** the hull and then the thruster surface.
- Select **Edit Surfaces** → **Intersect Surfaces**.

  This creates a curve on surface. Right now you won't trim the hull surface back. You will wait until later when you have other curves on surface; and therefore more trim regions to choose from.



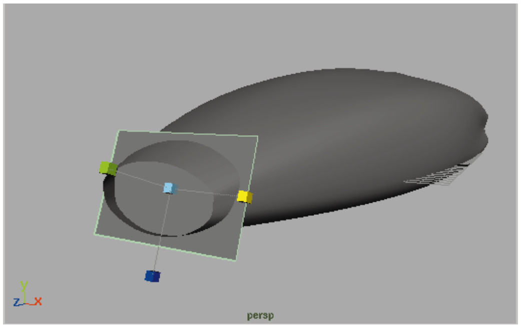
New curve on surface

A circular fillet

You will now create a circular fillet to define a softer look for the front of the ship. The fillet will be created between a primitive plane and the hull surface.

1 Create and place a primitive plane

- In the Perspective view, tumble until you see the front of the ship.
- Select **Create** → **NURBS Primitives** → **Plane**.
- **Rotate** the plane in **X** to **70** degrees**.**
- **Move**, and **Scale** the plane until it is intersecting the end of the ship body at a slight angle.
- Select **Edit** → **Delete by Type** → **History**.



New plane

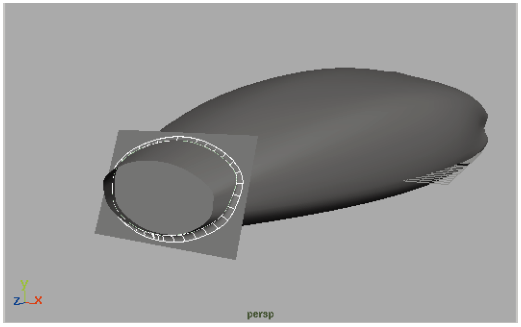
2 Fillet the two surfaces

- Press the **Shift** key and add the main ship body to the selection.
- Select **Edit Surfaces** → **Surface Fillet** → **Circular Fillet** - ❐ and set the following options:

  Create Curve On Surface is **On**;

  Radius to **0.5**.
- Click **Fillet** but do not close the option window.

  Oops! The fillet is going the wrong way. You need to redo the fillet so that it lies within the two surfaces.



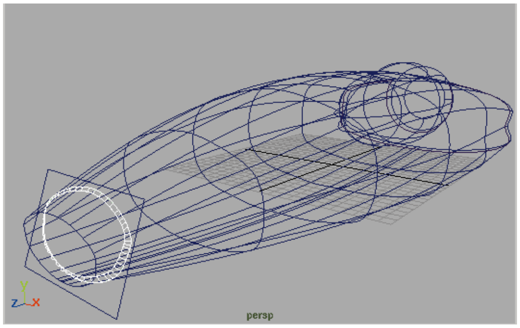
Fillet surface

- Press the **z** key to undo the fillet.
- In the Circular Fillet options window, set the following:

  Reverse Secondary Surface Normal to **On**.

  And if the fillet is in front of the plane, then also set:

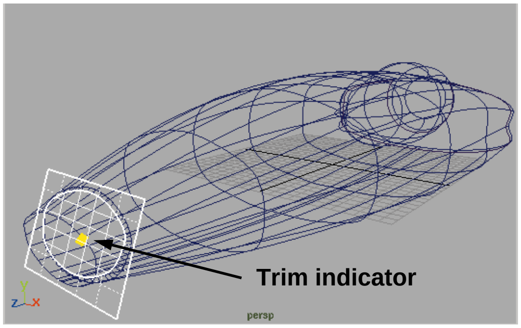
  Reverse Primary Surface Normal to **On**;
- Click **Fillet** then **Close**.
- Press the **4** key to go into wireframe mode where you can see the new filleted surface.



Corrected fillet

3 Trim the center piece

- **Select** the plane.
- Select **Edit Surfaces** → **Trim Tool.**
- Click in the middle area to place the trim indicator.



Trimmed region

- Press **Enter**.
- Select **Display** → **NURBS Smoothness** → **Custom**.
- Use the method taught earlier to edit the **Tessellation Criteria** in the Attribute Editor.

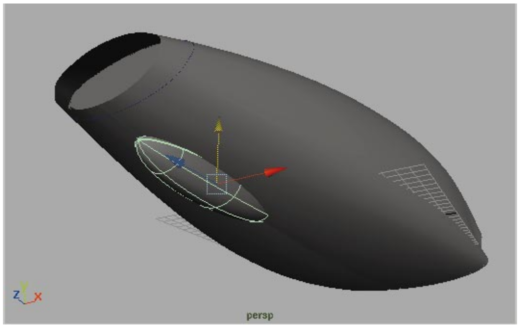
  Again, you will wait to trim the surface of the ship until all curves on surface have been placed.

The photon recess

To create an area for the ship to shoot photon blasts, you will create a recess at the bottom of the hull using a primitive sphere. You will intersect this shape and then trim both the surfaces.

1 Create and place a primitive sphere

- In the Perspective view, **Tumble** until the bottom of the ship is in view.
- Select **Create** → **NURBS Primitives** → **Sphere**.
- **Rotate** the sphere **90** degrees around the X-axis.
- **Move** and **Scale** the sphere to place it as shown below:



Primitive sphere

- Select **Edit** → **Delete by Type** → **History**.

2 Intersect the two shapes

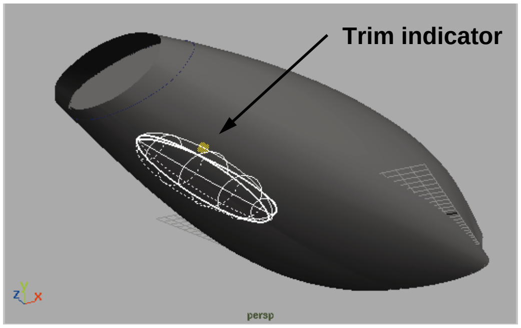
- Press the **Shift** key and add the ship's body to the selection.
- Select **Edit Surfaces** → **Intersect Surfaces** - ❐ and set the following options:

  Create Curves for to **Both Surfaces;**

  Curve Type to **Curve on Surface**.
- Click **Intersect** then **Close**.

3 Trim the sphere

- **Select** the sphere surface.
- Select **Edit Surfaces** → **Trim Tool.**
- Click on the area at the top of the sphere to place the trim indicator.

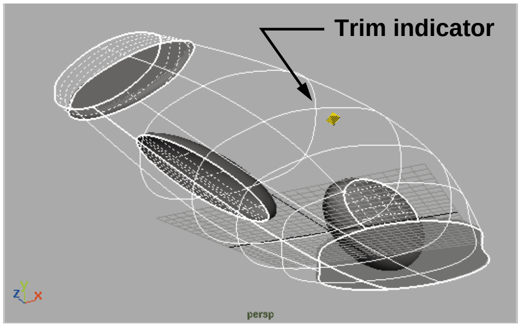


Trim indicator

- Press **Enter**.

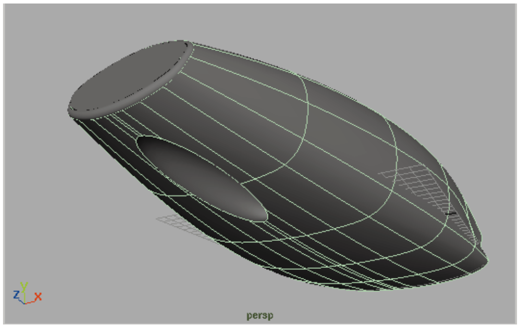
4 Trim the ship's body

- **Select** the ship's body surface.
- Press the **y** key to get the last tool.
- Click on the area in the middle of the surface to place the trim indicator.



Trim indicator

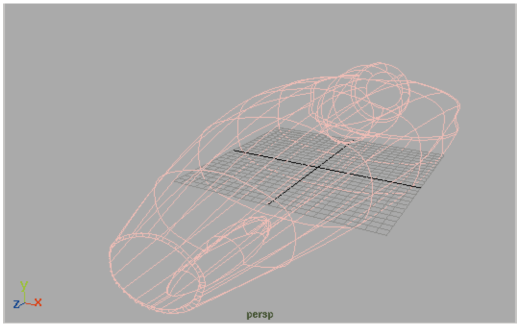
- Press **Enter**.
- Select **Edit** → **Delete by Type** → **History**.



Trimmed surfaces

5 Template the surfaces

- Select all of the surfaces.
- Assign the surfaces to the *mainBody* Layer.
- Template the *mainBody* layer.



Templated surfaces

6 Save your work

Building the wing

Like the main body, character curves will be created. This time the curves will act like the ribs of the surface which will then be connected using the loft tool. Construction history will then be used to tweak the surface by editing the curves.

1 Draw a primitive circle

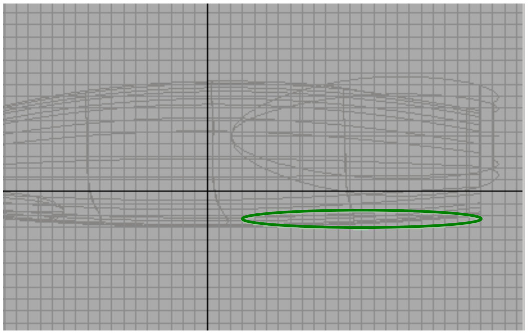
You will start with a primitive circle which is simply a premade curve.

- In the Side view, select **Show** → **All**.
- Select **Create** → **NURBS Primitives** → **Circle**.
- In the Channel box, set the following:

  Rotate Z to **90**;

  Scale Z to **12**;

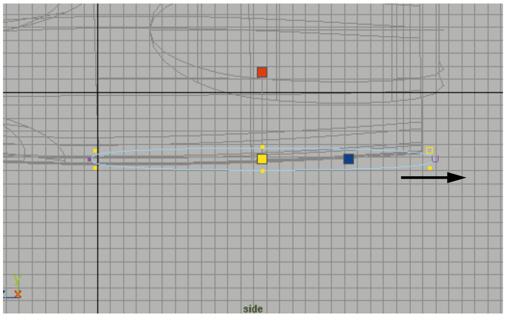
  Scale X to **0.85**.
- **Move** the circle to back of the ship as shown below:



Moved Circle profile curve

2 Reshape the circle

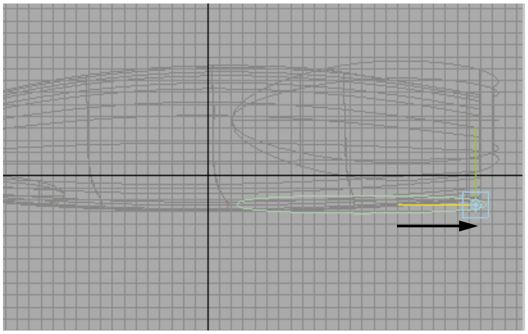
- Press **F8** to go into component selection mode.
- **Select** the middle 6 CVs.
- **Scale** the CVs out to push then closer to the edge of the profile to create a rounded square shape.



Scaled CVs

3 Move the profile's pivot location

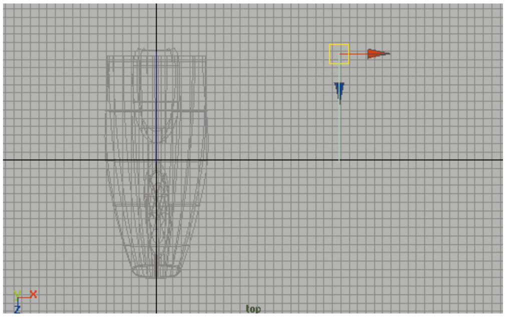
- Press **F8** to go back to object selection component mode.
- Press the **Insert** key to go into edit mode.
- **Move** the pivot along the Z-axis to the end of the profile.
- Press the **Insert** key to exit edit pivot mode.



Relocated pivot

4 Duplicate and move the new profile curve

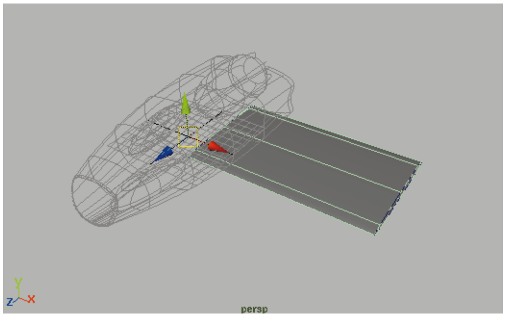
- Select Edit → **Duplicate**
- **Move** the new profile along the X-axis to about 2 times the width of the ship.



Repositioned profile curve

5 Loft the two profile curve

- Select both curves.
- Select **Surfaces → Loft.**

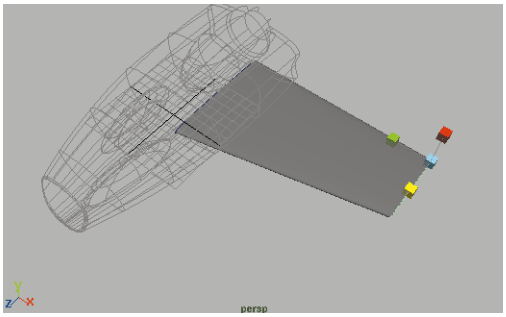


Lofted surface

6 Use construction history to modify surface

- Select the outside profile curve at end of the wing.
- **Scale** the curve in Z to **7.5.**

  With construction history, the surface will update as you are modifying the curve.



Modified surface

7 Rebuild the surface

You will be modifying the end of the wings even further. For more control, this will require more density of isoparms. You can achieve this by rebuilding the surface with more isoparms while retaining the shape.

- Select the *wing* surface.
- Select **Edit Surfaces → Rebuild Surfaces - ❐**, and set the following:

  Number of Spans U to **4**;

  Number of Spans V to **10**

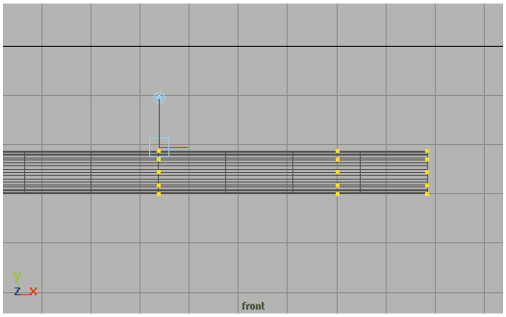
- Press **Rebuild.**
- Delete the two profile curves.

8 Bend the end of the wings

You will be modifying CVs by rotating them to curl the end of the wing upward.

- Select the *wing* surface.
- Press **F8** to go into component mode.
- Select the end three rows of CVs.
- Select the **Rotate** Tool.
- Press the **Insert** key to enter edit pivot point mode.

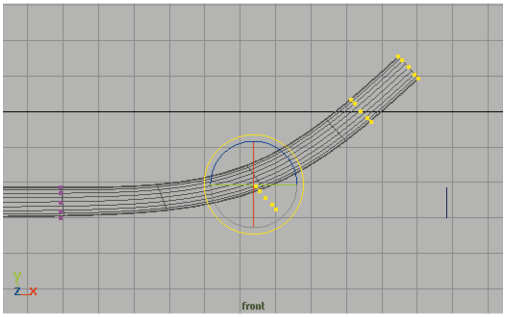
- In the front window, **Move** the pivot point in X until it is in line with the third row of CVs.
- **Move** the pivot point in Y until it is just above the top row of CVs.



Repositioned pivot point

9 Rotate the selected CVs

- Press **Insert** to exit edit mode.
- Rotate the CVs around Z until the end of the wing is pointing 45 degrees.



Rotated CVs

10 Deleting history

- Press **F8** to go back to object mode.
- Select the wing surface and choose **Edit → Delete by type → History**.

11 Duplicate and mirror the wing

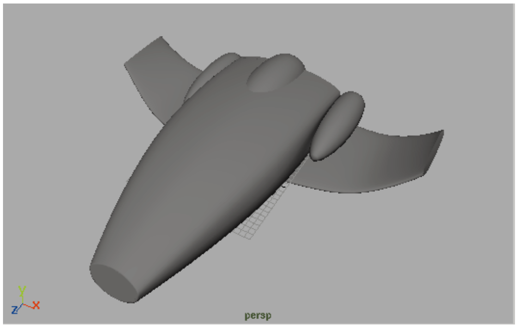
- Select the wing surface.
- Select **Edit → Duplicate**.
- **Scale X** to **-1** to mirror.

Details

To finish off the spaceship, you will add several details that will add to the look of the spaceship. To give you a chance to work on your own, you will build these elements yourself.

1 Create the wing thrusters

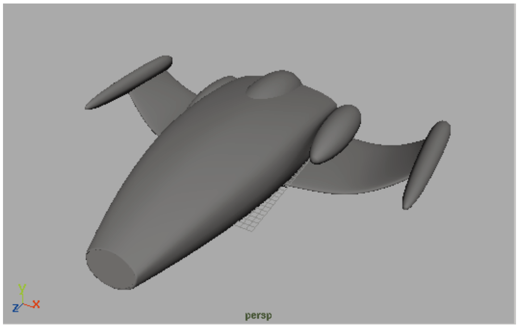
- Untemplate the mainBody layer
- Create two wing thrusters just above the wing. You can build these by either duplicating the existing thruster and scaling it down or by revolving new shapes from scratch.



New thrusters

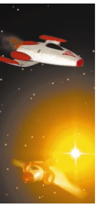
2 Add two wing torpedoes

- Create two wing torpedoes using either elongated spheres or revolved surfaces.



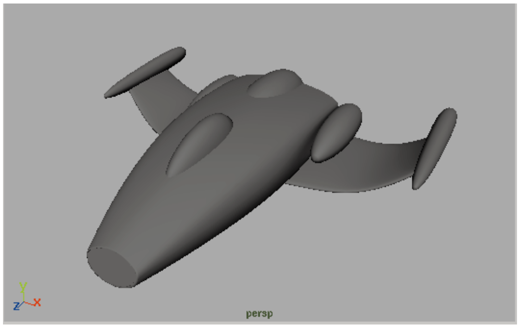
Wing torpedoes

3 Create a cockpit



- Create a cockpit surface in the center of the hull. You can use a half sphere and edit the CVs or you can revolve a profile.

  This surface does not have to be trimmed because it will properly intersect with the hull surface when rendered. Only use trimming when it is required visually.



Cockpit surface

Deleting history

To animate this spaceship, you will not be needing any of the construction history. While it is possible to maintain and even animate history, it is not required for this sequence. History can slow things down somewhat, and therefore should be deleted.

1 Delete history on the surfaces

Even though you have removed history from some of the surfaces, you can make sure by deleting history from all of the surfaces at the same time.

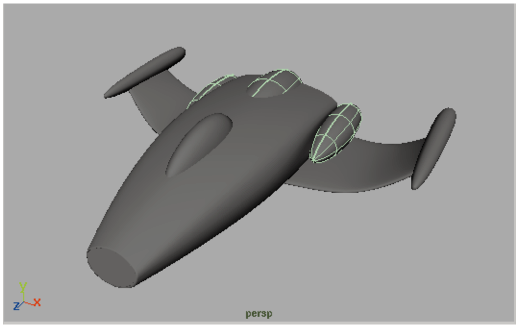
- Select all of the ship surfaces.
- Select **Edit** → **Delete by Type** → **History**.

Grouping the shapes

You will now organize the ship surfaces into a hierarchy. While only the root node will be used during the animation, it is a good idea to organize your work so that others can understand your models later on.

1 Group the thruster pieces

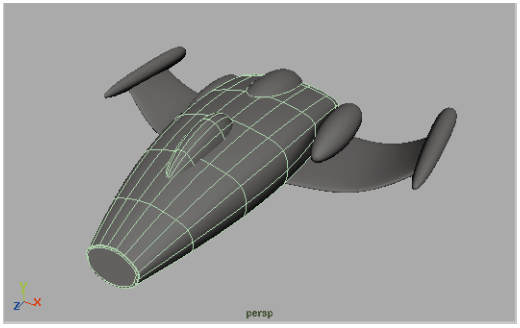
- **Select** all three of the thruster surfaces.
- Select **Edit** → **Group**.
- Rename the new group *thrusters* in the Channel box.



Thruster group

2 Group the hull pieces

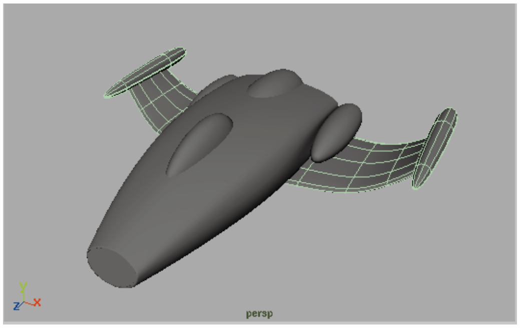
- **Select** the parts of the hull and cockpit excluding the thrusters.
- Select **Edit** → **Group**.
- Rename the new group *hull* in the Channel box.



Hull group

3 Group the wing pieces

- **Select** the two wing surfaces and the two torpedoes.
- Select **Edit** → **Group**.
- Rename the new group *wing* in the Channel box.

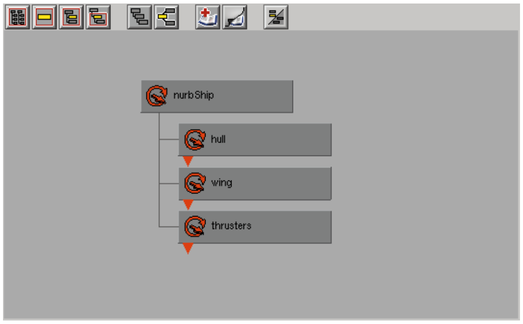


Wing group

4 Group the various groups

- Click on the **Select by hierarchy** button, then click on the **Select by hierarchy: Root** button (the fourth icon in the row).

  This will make sure that you are picking the top node of each group.
- Click-drag a selection box around the parts of the ship.
- Select **Edit** → **Group**.
- Rename the new group *nurbShip* in the Channel box.
- Open the Hypergraph to see the results. You can double-click on the group nodes to collapse them and make the graph easier to read.



Hierarchy view

5 Add a selection handle

- Make sure that the *nurbShip* node is selected.
- Select **Display** → **Object Components** → **Selection Handles**.

  This handle will make it easier to select the whole hierarchy later.

TEXTURING SURFACES

You are now going to texture the NURBS surfaces using projection maps. Projection maps use an external texture placement node to place the texture in relation to the surfaces. Your map will then be projected onto the surface based on your chosen projection technique.

Texturing the thrusters

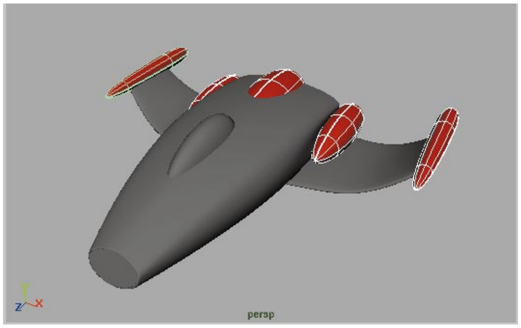
You will start by texturing the thrusters with a basic colored shading group.

1 Create a blinn shading group

- Open the Hypershade.
- Create a **Blinn** material node
- Press the **Show Downstream Connections** button to see the Shading Group node.
- Rename the Shading Group *thrustersSG.*
- Change the **Color** to red.

2 Assign it to the thruster surfaces

- Press **F8** to go back into object selection mode.
- **Select** the three thruster surfaces and the two torpedo surfaces.
- Select **Shading** → **Assign Shading Group** → **thrustersSG**.



First shading group assigned

Texturing the cockpit

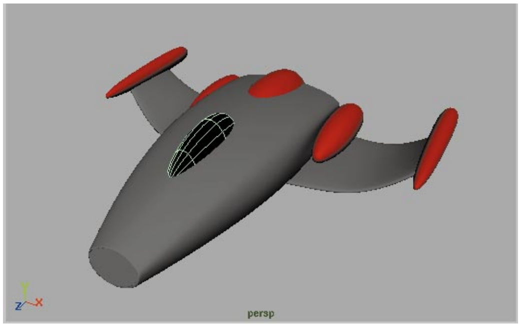
The cockpit will be textured using another simple color material.

1 Create a blinn shading group

- Create a **Blinn** material node, and name the Shading Group *cockpitSG*.
- Change the **Color** to black.

2 Assign it to the cockpit surfaces

- **Select** the cockpit surface.
- Select **Shading** → **Assign Shading Group** → **cockpitSG**.



Cockpit shading group assigned

Cylindrical projection map

To texture map the hull of the ship, you will use a cylindrical projection map. Because you want the texture to cover the main hull, the circular fillet and the front face a projection map is required.

A cylindrical projection is similar to wrapping a label around a tin can. The texture will be aligned with the length of the ship.

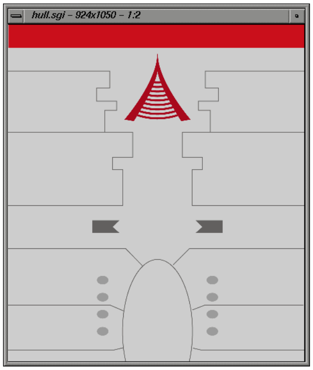
1 Create a blinn shading group

- Create a **Blinn** material node and name it *hullBlinn*.
- Name the Shading Group *hullSG*.

2 Apply a projected color map

- Open the Attribute Editor for the new Blinn Material node.
- Click on the **Map** button next to **Color**.
- In the 2D Textures section, click on the **As projection** option and click on the **File** button.
- Click on the *projection1* tab in the Attribute Editor.
- Set **Proj Type** to **Cylindrical**.
- Click on the *file1* tab.
- Click on the folder icon button next to **Image name**.
- Click on the file named *hull.sgi*, then click on the **See Image** button next to **Image name**.

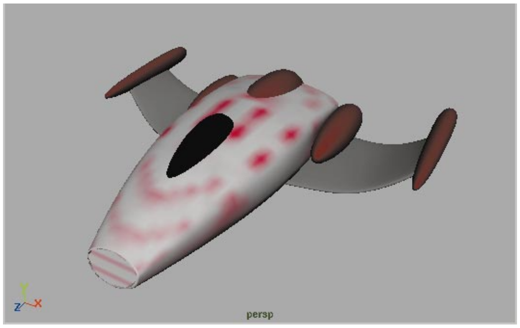
  This lets you see what the texture looks like in 2D.
- Click on the file named *hull.sgi*, then click on the **Open** button.



Texture map

3 Assign the shading group

- **Select** the main hull surface, the circular fillet surface and the front face of the ship.
- Select **Shading** → **Assign Shading Group** → **hullSG**.
- If you have hardware texturing, press the **6** key to preview the results. Press the **3** key to increase the surface display.



Shading group assigned

4 Position the projection icon

- Select **Shading** → **Assign Shading Group** → **hullSG - ❐**.

This opens the material node in the Attribute Editor.

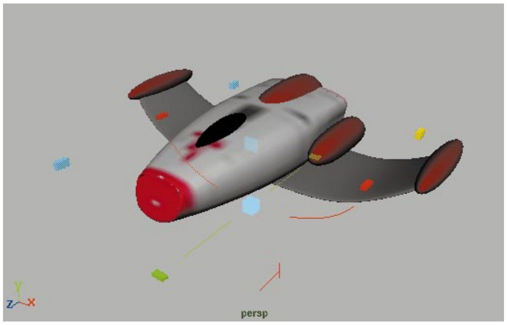
- Click on the Map button next to **Color**.
- Click on the *place3DTexture1* tab.
- Under **Transform Attributes**, set the following:

  Translate to **0, 0, 3**;

  Rotate to -**90, 0, 0**;

  Scale to **28, 28, 14**.

This places the projection icon so that it surrounds the spaceship surface. If it is not quite right for your model, tweak the values interactivley or enter different numbers.

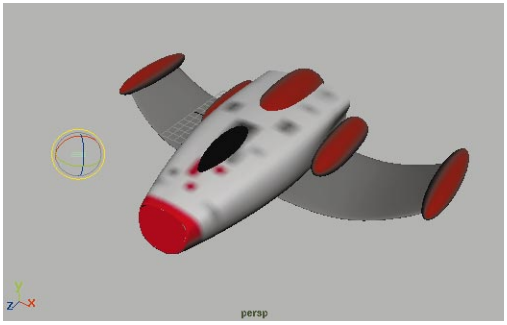


Texture placement icon

Note: You will notice that the placement icon has manipulator handles. For this lesson, the coordinate entry was used because the handles were too small.

5 Add a directional light

- Select **Lights** → **Create Directional Light**.
- **Rotate** the light until the light is looking down at the ship from the front.
- If you have hardware texturing press the **7** key.

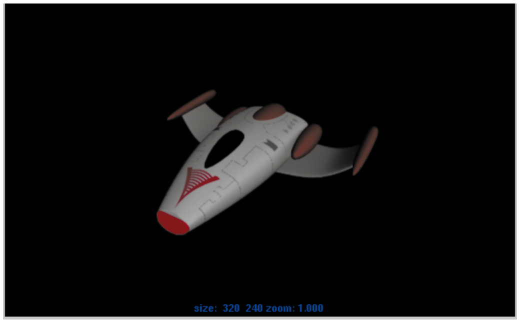


Directional light

6 Test render the surface

Now that you have a light, you can render the scene using software rendering.

- Open a Render View panel.
- In the Render View panel, select **Render** → **Render** → **persp**.



Rendered surface

7 Parent the 3D placement node

You will now parent the *place3dtexture* node to the hull surface so that it moves with the ship when it is animated later.

- In the Hypershade, Hypergraph, or Outliner, select the *place3dtexture* projection placement node.
- **Select** the main hull surface.
- Select **Edit** → **Parent**.

  Now the icon will move with the ship when it is transformed.

Planar projection map

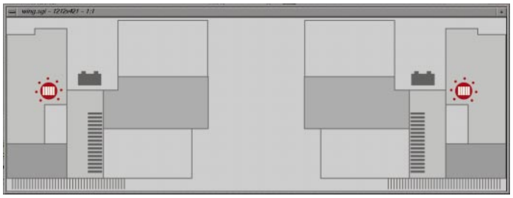
Another projection technique is planar mapping where the placement of the texture is a plane that is then projected down onto the surface.

1 Create and blinn shading group

- Create a **Blinn** material node and name it *wingBlinn*.

2 Apply a projected color map

- Press **Ctrl-a** to open the Attribute Editor.
- Click on the **Map** button next to **Color**.
- In the **2D Textures** section, click on the **As projection** option then click on the **File** button.
- Click on the *projection2* tab in the Attribute Editor.
- Set **Proj Type** to **Planar**.
- Click on the *file2* tab.
- Click on the **Folder** icon button next to **Image name**.
- Click on file named *wing.sgi* then click on the **See Image** button next to **Image name**.



Wing texture map

- Click on the **Open** button.

3 Assign the shading group

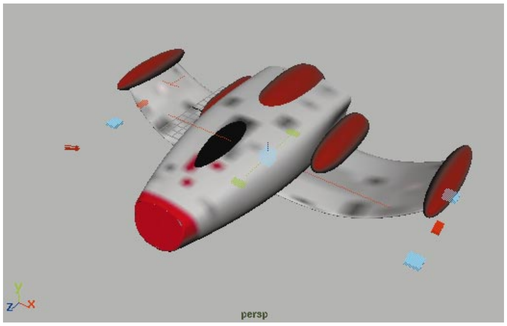
- In the perspective window, press **6** for shaded view.
- **Click-drag** with **MMB** the *wingBlinn* material node from the Hypershade to over the left wing in the perspective window and release the button.
- Repeat for the right wing.

4 Position the planar projection

- Open the Attribute Editor for *wingBlinn*.

  This opens the material node into the Attribute Editor.
- Click on the arrow icon button next to **Color**.
- Click on the *place3DTexture2* tab.
- Under **Transform Attributes**, set the following:

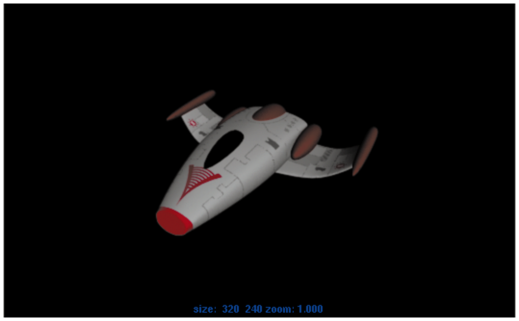
  Rotate to **90, 0, 0**.
- Click on the **Fit to group bbox** button in the **3D Texture Placement attributes** section.



Projection icon

5 Test render the surface

- Open a **Render View** panel.
- Select **Render** → **Render** →**persp**.



Rendered wings

6 Parent the place3DTexture node

You will now parent the placement node to the wing surface so that it moves with the ship when it is animated later.

- **Select** the *place3DTexture* node for *wingBlinn*.
- **Select** the wing surface.
- Select **Edit** → **Parent**.

  Now the icon will move with the ship when it is transformed.

7 Reuse textures as bump maps

- Use techniques taught in earlier lessons to reuse the *projection* node textures as bump maps. You can use either the Hypershade or the Hypergraph to help you.

8 Save your work

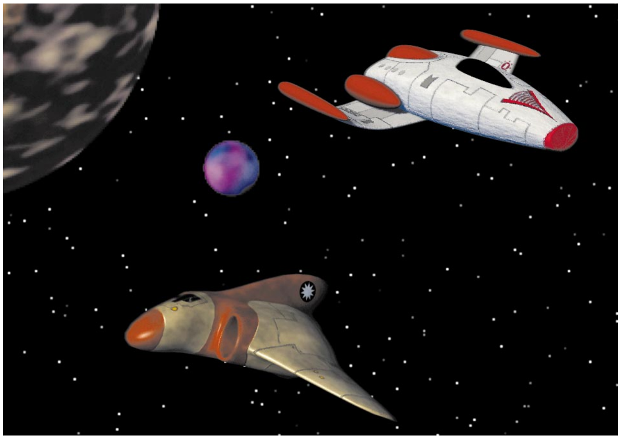
Conclusion

You now know how to begin working with NURBS modeling tools. You can lay down character curves and then use these to generate surfaces. You have also learned how to use construction history to edit your shapes after creation.

In the next lesson, you will animate the two ships along motion paths.

14 Animating the Ships

In this lesson, you will animate two spaceships using *path animation*. You will also add an image plane to the scene to help create a setting for the battle. The planets will be created using simple primitives. These ships will then be exported out so that they can be detailed in later lessons.



Space battle sequence

In this lesson you will learn the following:

- How to render a starfield with planets
- How to use the rendered space scene as an image plane
- How to create a reference to the ship files
- How to use path animation
- How to update the path markers
- How to shape the path to edit the animation

STARS AND PLANETS

To create a context for your spaceships, you will render a scene that contains a starfield and several planets. This simple space scene will then be rendered and used as an image plane backdrop for your duelling spaceships.

Initial set-up

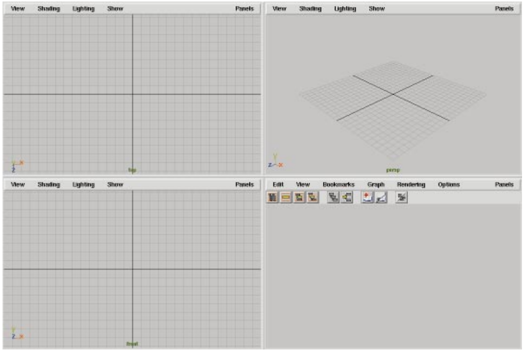
You are going to create a new file in which you will create the starfield image plane.

1 Start a new scene

- Make sure that your NURBS spaceship has been saved.
- Select **File → New Scene**

- Set up four panels containing the Perspective view, two orthographic views and the Hypergraph.

  In this lesson, the Hypergraph will be used to select various nodes of the shading groups that are to be created. This method of working with shading groups will be shown as an alternative to the Hypershade.


View Shading Lighting Show Panels
View Shading Lighting Show Panels
top
persp
View Shading Lighting Show Panels
Edit View Bookmarks Graph Rendering Options Panels
front

Suggested view pane layout for this lesson

Create the starfield

To create a starfield, you will use a procedural texture map as an image plane.

1 Set up the Perspective view

- Click on the Perspective view panel to make it the active panel.
- From this view panel, select **View → Select Camera**.
- Press **Ctrl-a** to open the Attribute Editor.
- In the **Camera Attributes** section of the Attribute Editor:

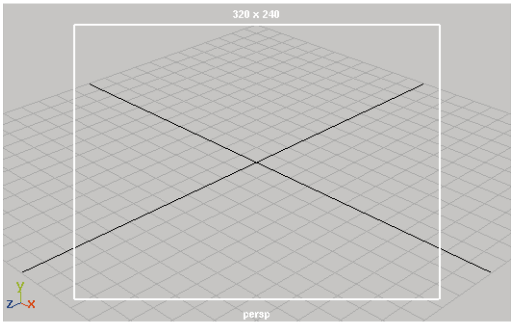
  Reduce the **Angle of View** to **20**.

  The narrow view angle will create less distortion in the shapes that appear in the view.
- In the **Film Back** section, set:

  Overscan to **1.4**.
- In the **Display Options** section, set:

  Display Resolution to **On**.

  You can now see what your final render size will be with some room around to help you position things.



Perspective view set up with overscan

2 Create an image plane

The starfield will be created as an image plane. Image planes are backdrops that stay behind the camera, as the camera moves around. You can place and render objects in front or behind the image plane. For this lesson, you want to place some planets in front of the plane.

- In the Attribute Editor, open the **Environment** attributes section.
- Click on the **Create** button next to **Image Plane.**

  This creates a new image plane node which is displayed as a yellow outline and cross lines in the Perspective view.
- In the **Placement** section for the new node, click on **Fit to Resolution Gate**.

  The image plane is now set to the same size and proportion as the rendering resolutions.

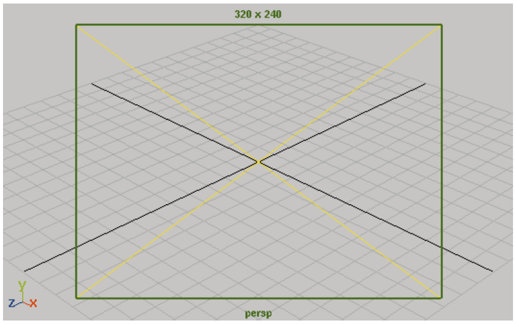
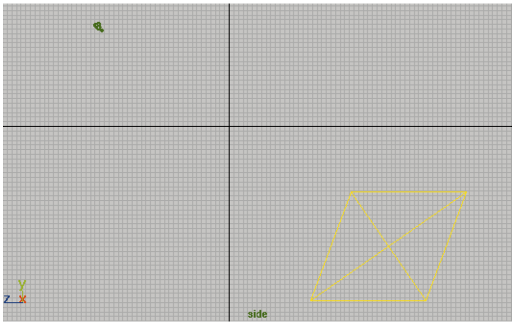


Image plane in Perspective view

The image plane is positioned in the 3D space of the scene and can be seen in the orthographic views. Its position in space is based on where the camera is looking.



Camera and attached image plane in side view

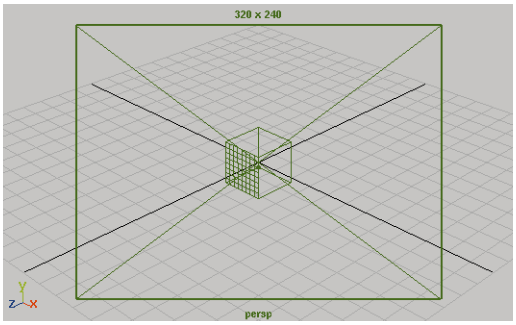
Note: Everything between the camera and the image plane will be rendered with the image plane as a backdrop.

3 Map a leather texture to the image plane

A 3D leather texture parameter can be used to create a star background.

- In the **Image Plane Attributes** section, click on **Map** next to **Texture**.
- In the **3D Textures** section, select the **Leather** texture.

  A 3D texture placement icon is positioned at the origin.



3D placement icon for leather texture

4 Edit the leather texture

The leather texture is edited to produce a random star pattern.

- In the **Leather Attributes** section, set:

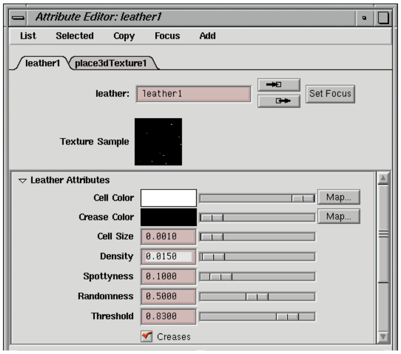
  Cell Color to white;

  Crease Color to black;

  Cell Size to **0.001**;

  Density to **0.015**.

  When you render later, these settings will create a starry pattern. Maya's procedural textures can often be used to produce your desired effect.



leather: leather1

Texture Sample

Leather Attributes

Cell Color
Crease Color
Cell Size 0.0010
Density 0.0150
Spottyness 0.1000
Randomness 0.5000
Threshold 0.8300
☑ Creases

Leather texture modified to produce the starfield

Creating three planets

To populate the space scene, you will need three planets placed around the starfield. These planets will act as the backdrop for the final scene.

1 Create three planets

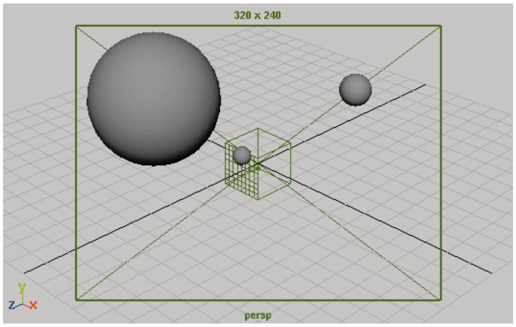
- Create 3 **Primitive NURBS Spheres**.
- Rename the spheres as follows:

  planet1

  planet2

  planet3
- **Scale** and **Move** the spheres as shown below.

  The relative size and position of the spheres in the 3D space of the scene is not important because they will be rendered into a 2D image plane later. Set them up so that their perceived relationships create the desired look in the Perspective view.



Spheres scaled and positioned as planets

Creating the planet textures

You will now texture the planets. You will use various procedural textures to refine a particular look, and then connect these to a shading group. In some cases, you will use a special window called the Connection Editor to make a direct connection between two attributes.

In previous lessons, you used the Hypershade to visualize how a shading group works. This time you will use the Hypergraph to work with these nodes.

1 Create and assign the first planet shading group

You will now create and assign a shading group using the Hypershade.

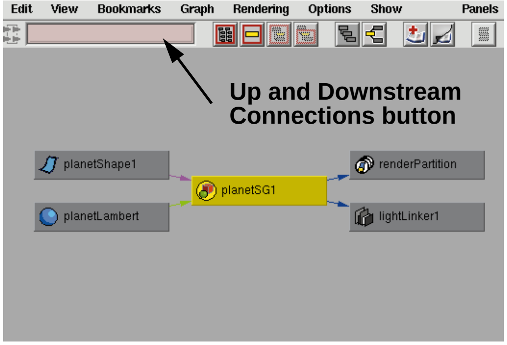
- Select **Window → Hypershade...**
- In the Hypershade, select **Create → Materials → Lambert.**
- Click in the Hypershade's work area to drop down the material.
- Rename the material node *planetLambert*
- **Click-drag** with **MMB** *planetLambert* onto *planet1.*

2 Display and rename the nodes

You can use the Hypergraph to see the shading group and material nodes.

- In the Hypergraph, select **Rendering → Show Shading Groups**.
- Select the *lambertSG* node.
- Click on the **Graph → Up and Downstream Connections** button.

  You can see the material node and the sphere surface connected to the shading group
- Rename the shading group node *planetSG.*



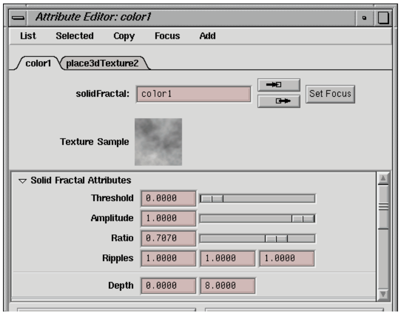
Shading group hierarchy

3 Map the material's color

To create the surface of the planet, you will use a solid fractal texture.

- **Select** the *planetLambert1* node.
- In the Attribute Editor, click on the **Map** button next to the **Color** attribute.
- In the **3D Texture** section, select a **Solid Fractal** node.
- Rename the texture node *color1.*

  The Solid Fractal Attributes can remain set to the defaults.



solidFractal: color1

Threshold 0.0000

Amplitude 1.0000

Ratio 0.7070

Ripples 1.0000 1.0000 1.0000

Depth 0.0000 8.0000

Solid fractal used as color map

4 Remap the color of the fractal

Currently, the fractal is only a grayscale image. You will use the Color Remap option to add color to the fractal.

- Open the **Effects** attributes section.
- Click on the **Insert** button next to **Color Remap**.

  A *RemapRamp* node is connected between the texture and the material nodes. The different positions on the ramp represent the grayscale values of the original texture. By changing the color on the ramp at the different values, you are setting a color on the original texture.

- In the **Ramp Attributes** section, set the ramp **Selected Colors** and **Selected Positions** as shown.

  Click on the top round icon and set the following:

  Selected Position to **0.65;**

  RGB to **0.57, 0, 0.22**.

  Click on the middle round icon and set the following:

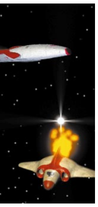
  Selected Position to **0.48;**

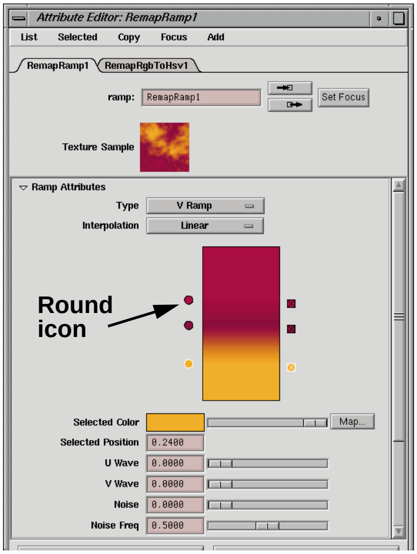
  RGB to **0.37, 0.01, 0.17**.

  Click on the bottom round icon and set the following:

  Selected Position to **0.24;**

  RGB to **1, 0.55, 0.17**.





ramp: RemapRamp1

Type: V Ramp

Interpolation: Linear

Selected Position: 0.2400

U Wave: 0.0000

V Wave: 0.0000

Noise: 0.0000

Noise Freq: 0.5000

Remap ramp of fractal

5 Map the material's bump

You now want to create a bump map node. This node will be used later to reconnect the color remap node as a bump.

- Open the *planetLambert* material node in the Attribute Editor.
- Click on the **Map** button next to **Bump Mapping**.
- In the Create Render Node window, open to the **Utilities** tab.
- Under **General Utilities**, select a **Bump 3d** node.
- In the **3d Bump Attributes** section:

  Set **Bump Depth** to **-1.0**

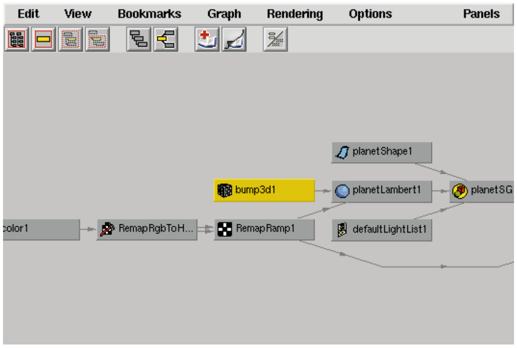
- In the **Effects** section:

  Set the **Bump Filter** to **0.5**.

6 Connect the remap node to the bump

To connect the remap node to the bump, you need to drag the texture onto the bump node. This will open the Connection Editor which lets you make the connection yourself.

- In the Hypergraph, drag with **MMB** the *RemapRamp1* node over the *bump3d1* node.



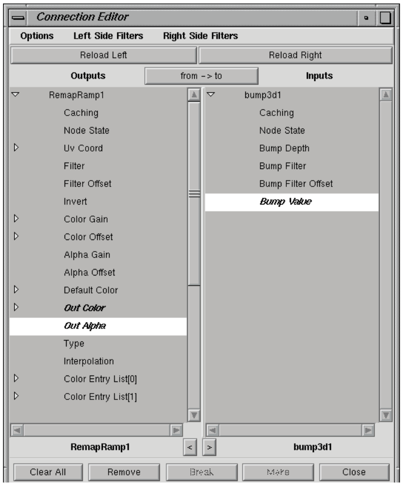
Connecting remap ramp node to 3D bump node

The Connection Editor opens. When Maya isn't sure which connection is most desired, it opens this window for you to complete the connection.

- Under **Outputs**, highlight **Out Alpha.**
- Under **Inputs**, highlight **Bump Value**.

This connects the *Remap of the Solid Fractal* node to the 3D bump for the sphere.

- Close the Connection Editor.



Connection Editor with connection to Bump Value

Note: The nodes are now connected in the Hypergraph. You can move your cursor over the arrow lying between the nodes, and the connected attributes will be highlighted.

7 Map the material's incandescence

- Open the attributes for the *planetLambert* material node.
- **Map** the **Incandescence** attribute with a **Solid Fractal** 3D Texture.
- Rename the texture node *clouds1*.
- In the **Solid Fractal Attributes** section set:

Threshold to **0.5**;

Amplitude to **0.8**;

Ratio to **1.0**;

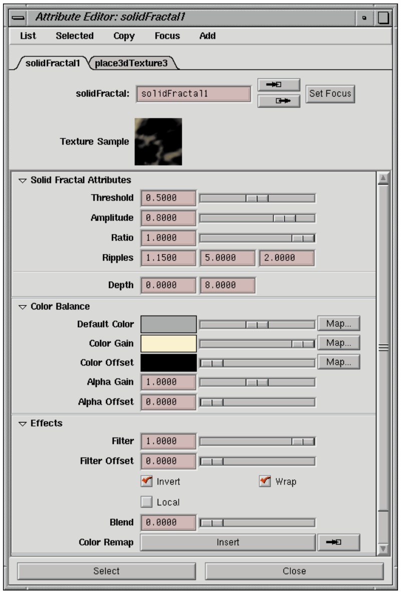
Ripples to **1.15**, **5.0**, **2.0**

- In the **Color Balance** section:

  Set **Color Gain** to a pale yellow.

- In the **Effects** section:

  Turn on **Invert**.



solidFractal: solidFractal1

Texture Sample

Solid Fractal Attributes
Threshold 0.5000
Amplitude 0.8000
Ratio 1.0000
Ripples 1.1500 5.0000 2.0000
Depth 0.0000 8.0000

Color Balance
Default Color
Color Gain
Color Offset
Alpha Gain 1.0000
Alpha Offset 0.0000

Effects
Filter 1.0000
Filter Offset 0.0000
☑ Invert ☑ Wrap
☐ Local
Blend 0.0000
Color Remap

Solid fractal settings for incandescence map

Create more planet shaders

To texture map the remaining planets, you will duplicate the first shading group, then edit the various texture nodes to produce different effects.

1 Duplicate the planet's shading group

- In the Hypergraph, **Select** the *planetSG1* node.
- From the main menu, select **Edit → Duplicate - ❐**, and in the Duplicate Option window:

  Turn on **Duplicate Upstream Graph**;

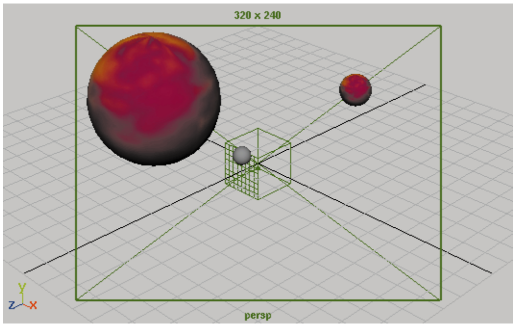
  Click on **Duplicate**.

A duplicate copy of the network of nodes that make up the shading group is made.

Tip: This is the same as selecting **Duplicate → Shading Network** from the Hypershade's **Edit** menu.

2 Assign the new Shading Group

- **Click-drag** with **MMB** *planetSG1* from the Hypergraph onto *Planet2* in the view panel.



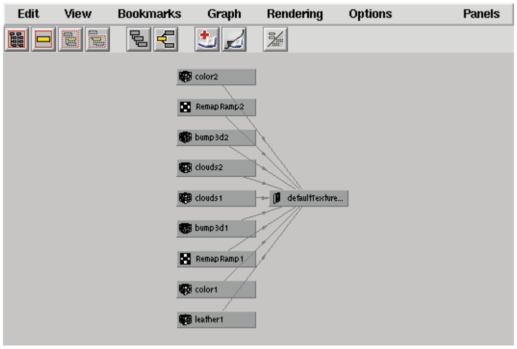
Duplicate shading group assigned to planet2

3 Change the color of planet2

Modifying the fractal, as seen in the **Texture Sample**, will create different planet images.

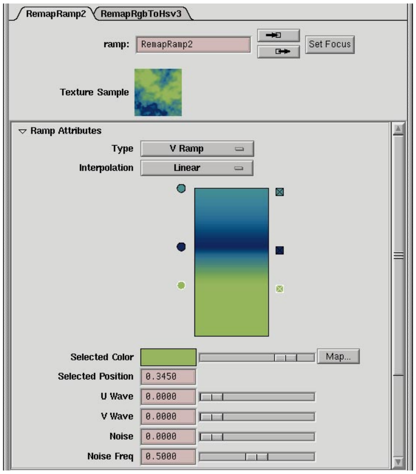
- In the Hypergraph, select **Rendering → Show Textures**.

All of the texture nodes in the scene are displayed. Notice that all of the texture nodes for the planet shading group have been duplicated.



Hypergraph view of the texture nodes in the scene

- **Select** the *color2* node.
- In the Attribute Editor, change the values for **Threshold**, **Amplitude** and **Ratio** to whatever you like.
- **Select** the *RemapRamp2* node.
- Change the **Colors** and **Positions** to produce a new map.



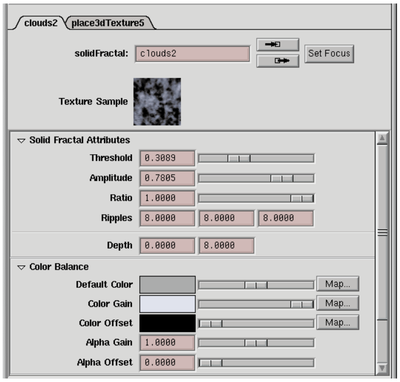
ramp: RemapRamp2
Texture Sample
Ramp Attributes
Type: V Ramp
Interpolation: Linear
Selected Color
Selected Position: 0.3450
U Wave: 0.0000
V Wave: 0.0000
Noise: 0.0000
Noise Freq: 0.5000

Fractal and remap nodes modified for planet2

Tip: Remember that the value of the remap colors will be used by the bump map.

4 Change the clouds for planet2

- **Select** the *clouds2* node.
- Change the **Threshold**, **Amplitude**, **Ratio** and **Ripples** attributes to produce new cloud formations.
- Change the **Color Gain** to a different color.



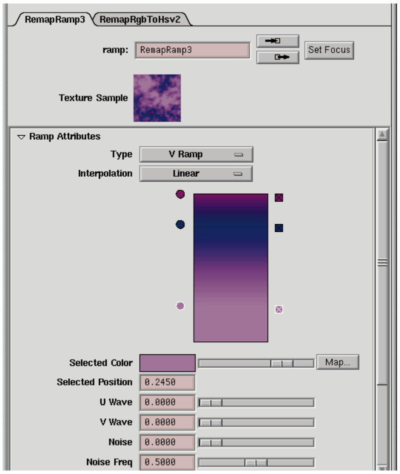
solidFractal: clouds2
Texture Sample
Solid Fractal Attributes
Threshold: 0.3089
Amplitude: 0.7805
Ratio: 1.0000
Ripples: 8.0000 8.0000 8.0000
Depth: 0.0000 8.0000
Color Balance
Default Color
Color Gain
Color Offset
Alpha Gain: 1.0000
Alpha Offset: 0.0000

Fractal cloud node modified for planet2

Tip: The fractal clouds for the other planets should have more detail to account for the smaller size. This could also be done by adjusting the scale of the 3D texture placement icon.

5 Modify the shading group for planet3

- Repeat the steps outlined above to create the shader appearance of *planet3*.



ramp: RemapRamp3

Ramp Attributes

Type: V Ramp

Interpolation: Linear

Selected Position: 0.2450

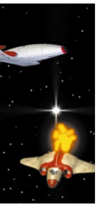
U Wave: 0.0000

V Wave: 0.0000

Noise: 0.0000

Noise Freq: 0.5000

Fractal and remap nodes modified for planet3



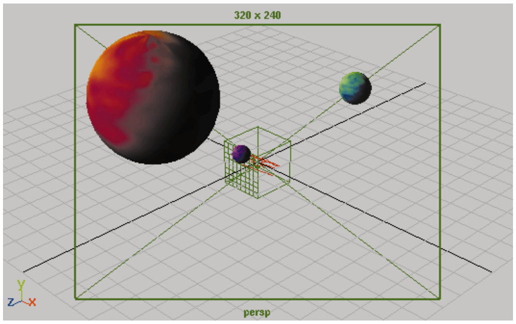
Render the scene

You will now render the space scene to create a bitmap image. This is the image that you will use later as an image plane in the final battle scene.

1 Place lights in the scene

- Create a **Directional Light**.
- **Rotate** the light so that the direction is coming from the left of the view and from above.
- In the Attribute Editor, set the **Color** to a pale yellow.
- Create an **Ambient Light**.
- Set the **Color**, **Intensity** and **Ambient Shade** attributes.

  Both lights should approximately match the lights created for the polyShip scene.



Hardware rendering of positioned lights

Note: When adding lights, plan ahead to determine how the lights will be positioned in the final scene. Keep in mind that they should be consistent with the lighting of the background image.

2 Increase sphere tessellation

- **Select** the sphere for the largest planet.
- Open the **Tessellation** section of the Attribute Editor.
- Set the **Display Render Tessellation button** to **On**.
- Set the following:

  Set **Curvature Tolerance** to **Highest Quality**

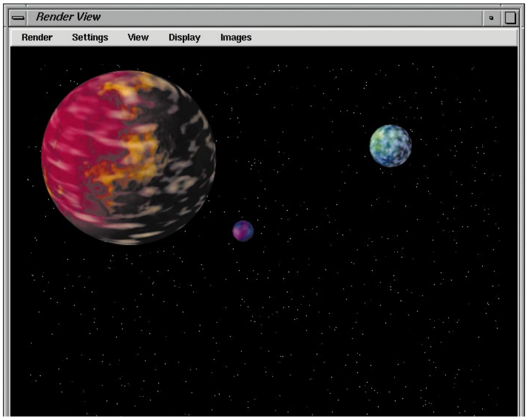
  Set **U Divisions Factor** to **2**;

  Set **V Divisions Factor** to **2**;
- Set the **Display Render Tessellation button** to **Off**.

  Depending on their size in relation to the overall view, you may want to increase the tessellation of the other two spheres.

3 Render the background image

- Select **Window → Render Globals...**
- In the Resolution section set the **Render Resolution** to **640x480**
- In the Anti-Aliasing Quality section, set the **Presets** to **Production Quality**.
- Open the Render View window.
- Select **Render → Render → Persp.**



Rendered background image

4 Save the rendering

- From the Render View window, select **File → Save Image...**
- Name the image *background* and save it into the *sourceimages* directory.

5 Save your Maya file

- **Save** the file as *planets* in the *scenes* directory.

THE SPACESHIP SCENE

You will now start to create and animate the duelling spaceship scene. You will start with the polygonal spaceship from the last lesson. Then you will animate a proxy object for the NURBS spaceship. Lastly, you will import and animate the background image as an image plane.

Creating a reference

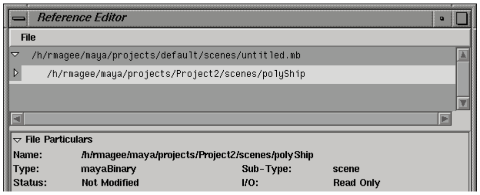
You are now going to start a new file, then create a reference to the first spaceship. By referencing the file, you begin to create a situation where files can be shared by multiple users.

1 Create a new file

- Make sure that your *planets* file has been saved then select **File → New**.

2 Create a reference to the polyShip

- Select **File → Create Reference...**
- Select the *polyShip* file from the file window then click **Reference**.
- Select **File → Reference Editor...**
- Click on the arrow to see that the new file is a reference in the new scene.



```
▽ /h/rmagee/maya/projects/default/scenes/untitled.mb
▷    /h/rmagee/maya/projects/Project2/scenes/polyShip
```

▽ File Particulars

Name:	/h/rmagee/maya/projects/Project2/scenes/polyShip		
Type:	mayaBinary	Sub-Type:	scene
Status:	Not Modified	I/O:	Read Only

Reference Editor showing polyShip file

- Close this window.

Path animation

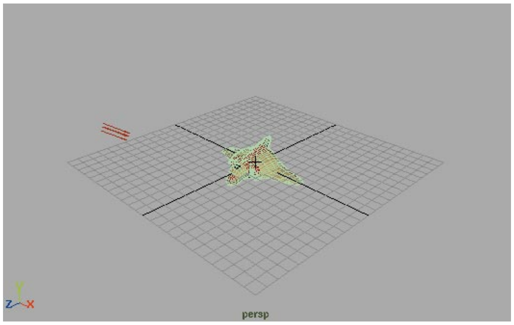
Path animations are created by assigning an object or series of objects to a path. This creates a special *motionPath* node that allows you to key its motion along the path.

1 Scale the spaceship

- **Select** the polyShip surface. Be sure not to select either of the lights that were referenced with the ship.

  In the Channel box, you will see that the name of the ship is *polyShip_polyShip*. This is because the file's name has been used as a prefix for all referenced nodes. This helps recognize referenced nodes and prevents name conflicts.

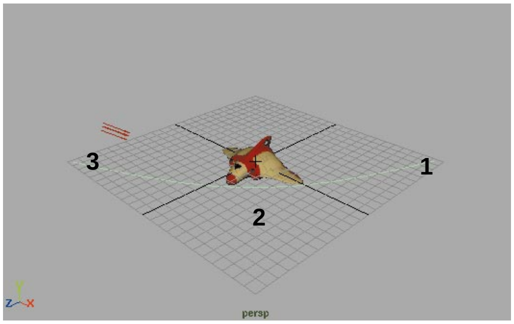
- Select **Edit → Group**.
- In the Channel box, rename the group as *polyShipAnim*.
- Select **Display → Object Components → Selection Handles**.
- **Scale** the selected node to **0.3** in all directions.



Imported spaceship

2 Draw a path animation curve

- Select **Create → EP Curve Tool**.
- Click three times to draw a curve diagonally across the ground grid then press the **Enter** key.



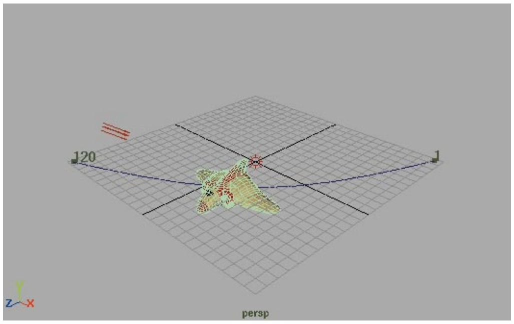
Path curve

3 Attach the ship to the path

- Change the **Time slider** range to **120** frames.
- **Select** the *polyShipAnim* node using its selection handle and then press the **Shift** key and select the path.

Note: In order to create a path animation, the path must be picked last. The last object picked is indicated in green.

- Go to the **Animation** menu set.
- Select **Animate → Paths → Attach to Path**.
- Play back the results.



Ship attached to path

Note: If you want to preview the correct playback timing, remember to set your playback speed to normal in the General Preferences window.

4 Edit the motion path input node

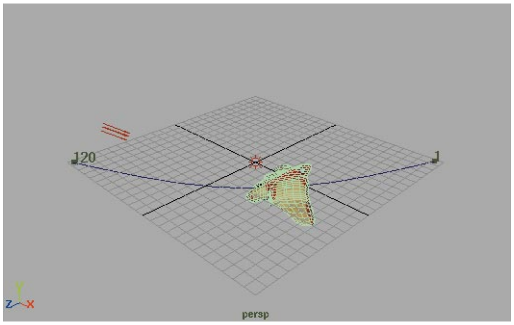
The ship is moving down the path but it is not aimed in the correct direction. You can change this using the *motionPath* input node.

- In the Channel box, click on the *motionPath* input node.
- Press **Ctrl-a** to open the Attribute Editor. In this window, set the following:

  Follow to **On**;

  Front Axis to **Z**;

  Up Axis to **Y**.
- Close this window.
- Play back the results.

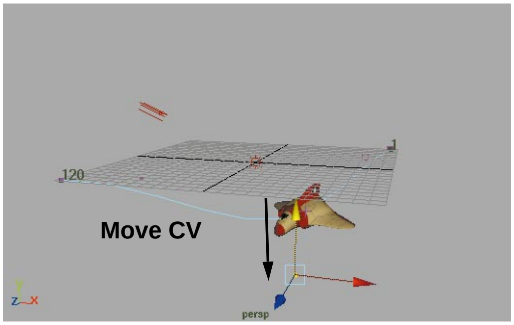


Spaceship with follow

5 Edit the path's shape

You can edit the shape of the path using the curve's control vertices and the object will follow the path.

- **Select** the path curve.
- Press the **F8** key to go into component select mode.
- **Select** the middle CV.
- **Move** it down along the Y-axis.



Edited path curve

- Press the **F8** key to go into object select mode.
- Play back the results

Tip: You can press the **Alt** key and tumble in the Perspective window as the animation is playing back. This lets you preview the shape of the path from different angles.

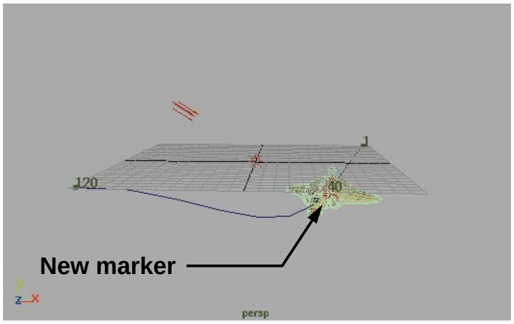
6 Key the path's U value

You can set a key on the *motionPath's* U value to add more markers to the path.

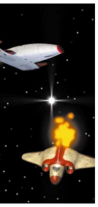
- Go to frame **40**.
- **Select** the *polyShipAnim* node using its selection handle.
- In the Channel box, click on the *motionPath* input node.
- In the Channel box, click on the **U Value** channel name to highlight.

- With your right mouse button, select **Key Selected** from the pop-up menu.

  A new marker is placed where the new key is set. You are setting a key on the position of the ship along the U direction of the curve. The value represents the parameter of the curve.



New path markers

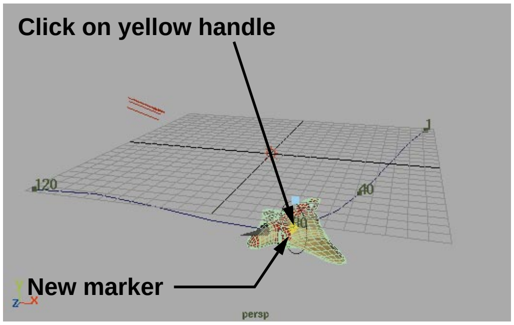


7 Key a second value

This time you will key the U value using the show manipulator tool and Auto Key.

- Go to frame **80**.
- Select the **Show Manipulator** tool.

  A manipulator appears with handles for positioning the object along the path and twisting. You will use the middle handle to move the ship along the path.
- Click on the **Auto Key** button at the right end of the Time slider to turn it **On**.
- Click-drag on the yellow manipulator handle to drag the ship back. Be sure not to click-drag on the arrow.
- Another path marker is placed on the curve.



New path marker

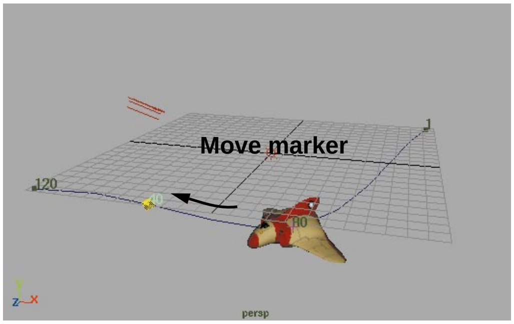
Tip: It is always good to remember that input nodes may have manipulators which you can access using the show manipulator tool.

8 Edit the path marker's position

The position of the markers can be moved to edit the animation of the ship.

- Click on the **Auto Key** button to turn it **Off**.
- **Select** the path marker that is labeled as 40. Click on the number without touching the curve to select the marker on its own. It will be colored yellow when selected.
- Select the **Move** tool.
- **Move** the marker past the marker labeled as 80.

  The marker is constrained to the curve as you move it.



Edited path curve

- Play back the results.

  The ship animates up to the 40 marker, then goes backwards to the 80 marker, and then forward to the end of the curve. The position of the ship at these key frames can be set using the markers.

9 Edit the timing

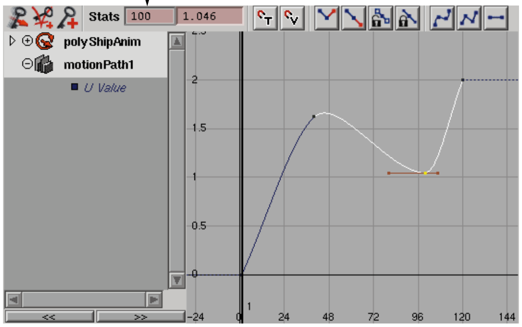
Since the marker points are simply keys set on the U Value of the *motionPath* node, you can edit the timing of the keys in the Graph Editor.

- **Select** the ship using its selection handle, then click on the *motionPath* input node in the Channel box.
- Open a Graph Editor panel.
- Move your cursor into the Graph Editor window and press **a** to frame all into the panel.

  The position of the attached object in the U direction of the curve is mapped against time. You can see that a key has been set for each of the path markers.
- **Select** the key at frame **80**.
- In the Graph Editor's Stats area change the time from **80** to **100**.
- In the Graph Editor, select **Tangents → Flat**.

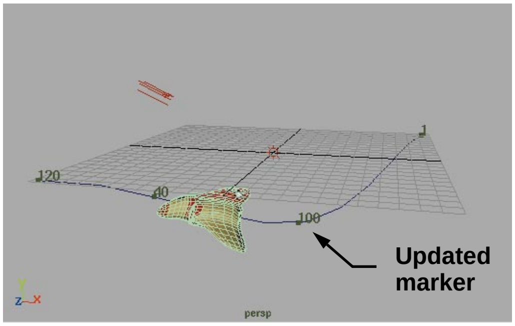
  You can edit the effect of the in-between frames for path keys using the same techniques as you have used for set keys.

Edit this field



Edited path curve

You can see that the path marker is now labeled as 100 in the view panel.



Edited path curve

10 Remove the middle two markers

While the markers have shown you how to make the ship move back and forth, this is not how you want the spaceship to animate. You will remove the middle markers to return to a two marker path.

- **Select** the markers that are labeled as **40** and **100**. Make sure that no other objects are selected.

Tip: You can select the keyframes from the Graph Editor to delete them as well.

- Press the **Backspace** key on your keyboard.

Animate a second ship

You will now animate the NURBS spaceship along a second path.

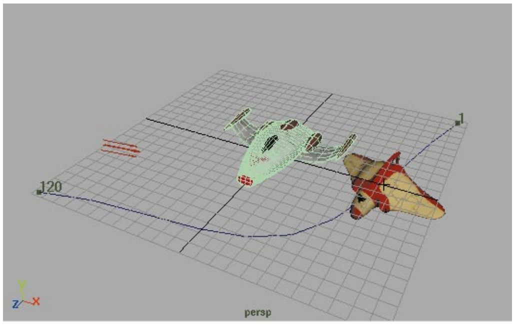
1 Create a reference to the nurbShip

- Select **File → Create Reference**.
- **Select** the *nurbShip* file from the file window.

  Now this file is also referenced into the scene. Just like the polyShip, it is also too big right now.

2 Scale the NURBS ship down

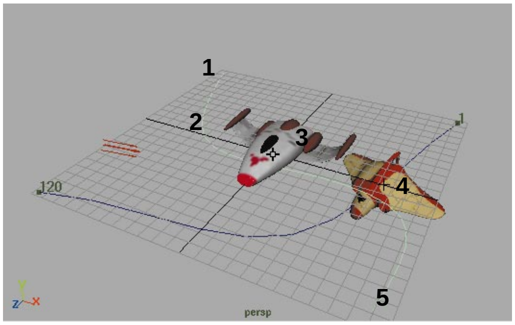
- **Select** the nurbShip using its selection handle.
- **Scale** the ship down by **0.15** in all directions.



NURBS spaceship scaled down

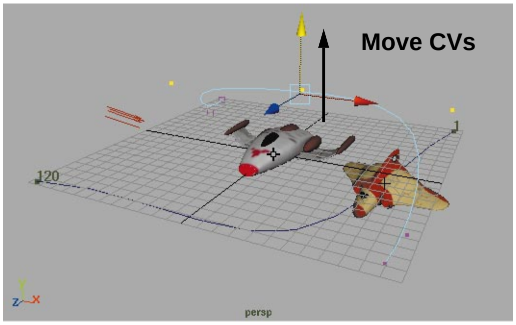
3 Draw a second path curve

- Select **Create → EP Curve Tool**.
- Click five times to draw a curve that snakes diagonally across the ground plane in the opposite direction from the first curve. Press the **Enter** key.



New path curve

- Press **F8** to go into component select mode.
- **Select** the middle CVs and **Move** them up so that this curve passes over the first curve.



Edited path curve

- Press **F8** to go back to object select mode.

4 Attach the ship to the path

- Go to the **Animation** menu set.
- **Select** the *nurbShip* using its selection handle.
- Press the **Shift** key and **Select** the path curve. Remember that the path must be selected last.
- Select **Paths → Attach to Path -** ❏.
- In the option window set:

  Follow to **On**;

  Front Axis to **Z**;

  Up Axis to **Y**;

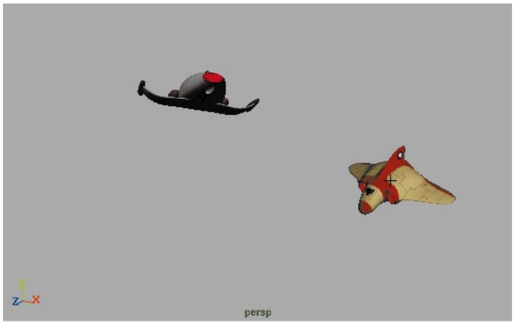
  Bank to **On**.

- Click **Attach** then **Close**.
- Select **Display → Grid** to turn off the ground grid.
- From the Perspective view panel, select **Show → Curves** and **Show → Lights** to turn these object types off.

Now you can preview what the ships look like on their own.

- Play back the results.

With banking turned on, the nurbShip tilts to the side as it grounds the corners of the path.



NURBS spaceship with banking

5 Save your work

- From the **File** menu, select **Save Scene As...**
- Enter the name *spaceScene* next to the file's path.
- Press the **Save** button or press the **Enter** key.

Setting up the image plane

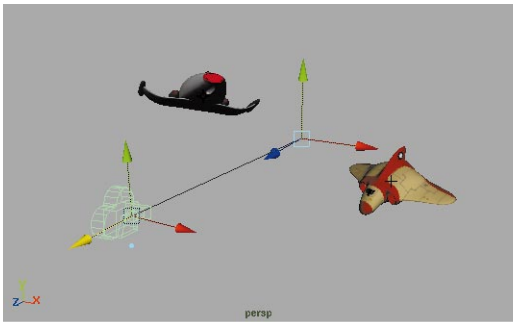
You are now going to import the background image plane. This plane will be used behind the spaceships to help compose the final shot. Once it is in place, you will edit the paths to suit the final scene.

1 Create a new camera

You will create a new camera to hold the image plane.

- Select **Create → Camera**.
- This places a camera at the origin.

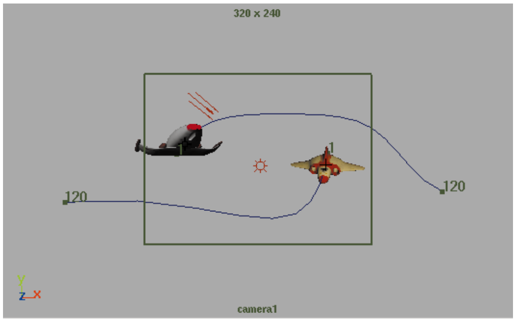
- Select **Show Manipulator** tool.
- Edit the look at and the eye points of this camera to look at the ships from the front.



New camera placement

2 Set up your panels

- In one of the orthographic panels, select **Panels → Perspective → camera1**.
- Press **5** or **6** to shade the *camera1* panel.
- Dolly, tumble and track in the *camera1* panel until you can see both spaceships.
- From the *camera1* view panel, select **View → Camera Settings → Resolution Gate**.



Resolution gate

3 Add the image plane

- From the *camera1* view panel, select **View → Select Camera**.

  This selects the nodes of the camera.
- Press **Ctrl a** to open the Attribute Editor.
- Open the **Environment** section of the *cameraShape1* node, then click on the **Create** button next to **Image Plane**.
- In the image plane node, click on the **Folder** button next to **Image Name**.
- Choose the *background* image file from the *sourceimages* directory.

  This is the image file that you rendered earlier in the lesson.

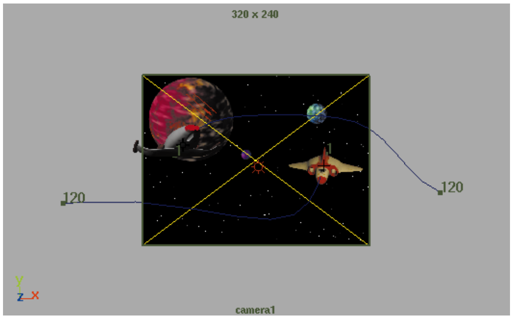


Image plane in place

4 Set up the image plane

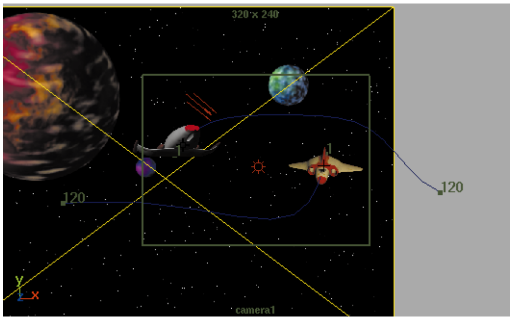
- In the **Placement** section, set the following:

  Size to **2.83**, **1.9**;

  Depth to **500**;

  Offset to **-0.5**, **-0.1**.

  The size setting doubles the size of the image in relation to the resolution gate, the depth setting moves it away from the camera and the offset changes the position in relation to the film back.



New image plane placement

5 Key the image plane

- In the **Time slider**, go to frame **1**.
- In the **Placement** section of the Attribute Editor, click on the name **Offset** with the right mouse button and select **Set Key**.

  The two offset fields should turn green to indicate that they have been keyed.
- In the **Time slider**, go to frame **120**.
- In the **Placement** section, set the following:

  Offset to **0.1**, **0.1**.
- Again, click on the name **Offset** with the right mouse button and select **Set Key**.
- Play back the results.

  The image plane is now animating from one position to another. This will give the impression that the camera is moving in space.

6 Save your work

Edit the path curves

You are now going to edit the animation of the two ships to create the final sequence.

1 Edit the curve shape for both curves

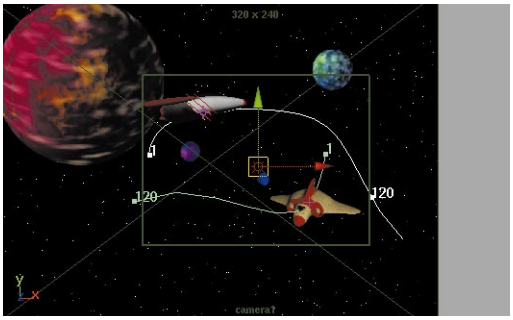
- Edit the CVs on the two path curves to create an animation where the nurbShip moves up and over the polyShip.

  You can set the shape of the curve to anything you desire. The key is to make sure that at frame 50, the nurbShip is above the polyShip so that it can shoot a photon torpedo in the next lesson.

Tip: Changing the **display mode** of the Image Plane can make it easier to see the objects in your scene when working.

2 Edit the position marker

If required, change the position of the path markers to refine the animation along the paths.



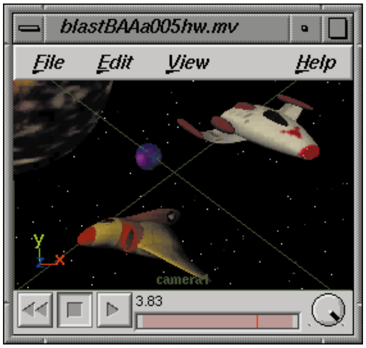
Updated paths

3 Save your work

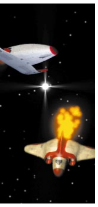
4 Playblast the final sequence

- In the *camera1* view panel, select **View → Camera Settings → No Gate**.
- Select **Window → Playblast**.

Play back the sequence using the Movie Player controls.



Playblast preview



Conclusion

You now have the two ships animated along two paths. In the next lesson, you will add visual effects to the scene as the nurbShip shoots photon torpedoes at the polyShip. You will then render the scene with the effects in place.

15 Visual Effects

In this lesson, you will add visual effects to your spaceship scene. The first few effects will use Maya's Digital OptiF/X capabilities. These effects include the photon blasts that are being shot at the polyShip, the photon's explosion and the nurbShip's thruster flame.

Visual effects

First you will add a particle thruster flame and then you will add smoke to the polyShip's wing after it is grazed by the OptiF/X explosion.

In this lesson you will learn the following:

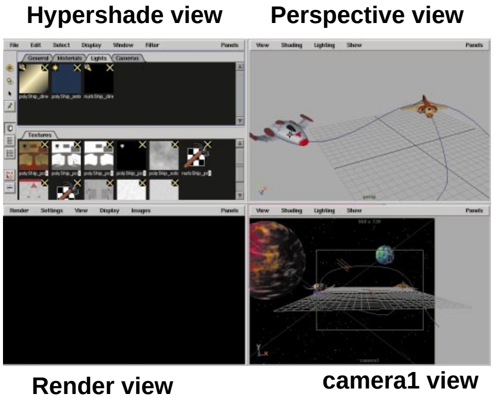
- How to create an OptiF/X fog light thruster
- How to create an OptiF/X photon torpedo
- How to create an OptiF/X explosion
- How to create a particle-based thruster for the polyShip
- How to add particle smoke to the polyShip wing

Initial set-up

You will continue this lesson using the space scene saved in the last lesson. You will add visual effects to the two animated spaceships. You will set keys on these effects in order to integrate them with the existing sequence.

1 Set up your view panels

- Create four view panels.
- Place the panels as shown below.
- In the Perspective view, select **Show → Cameras** to turn off the display of *camera1* and its image plane.



View panel layout

In the Hypershade, you will see all the shaders belonging to the scene. You may remember that some of these nodes are part of referenced files – in this case, the nurbShip and polyShip files. You can edit attributes for these shaders and assign them to other objects if required. Just remember that the original shader is part of another file.

CREATING OPTIF/X

OptiF/X are visual effects that you can add to your lights. These include, fog, glows and halos. Each of these effects generates nodes with attributes that you can animate.

To start, you will create a thruster for the nurbShip using a fog light. This light will use an animated fractal texture to give it a more interesting look. You then use animated glows to create a photon torpedo and an explosion.

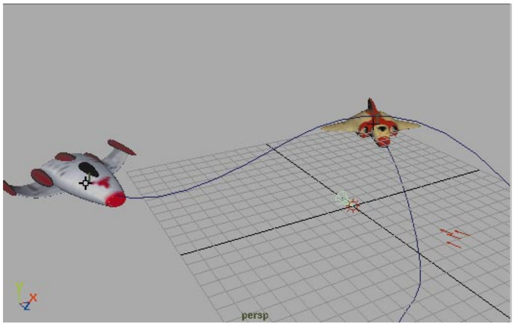
Building a thruster light

The thruster flame is going to be built using a spot light. You want the light to be positioned on the nurbShip so that it is pointing out of the rear thruster.

1 Place spot light in scene

- Go to the **Rendering** menu set.
- Select **Lights → Create Spot Light.**

  This places a spot light at the origin.



New light at the origin

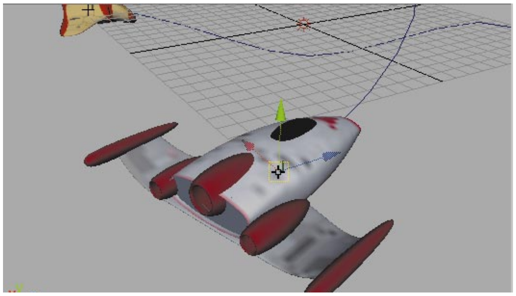
2 Parent the light into the nurbShip

- Press the **Shift** key and select the nurbShip using its selection handle.

- Select **Edit** → **Parent** - ❐ and set the following;

  Preserve Position to **Off**.

- Click on **Parent** then **Close**.
- Select the **Move** tool.



Parented light with global move manipulator

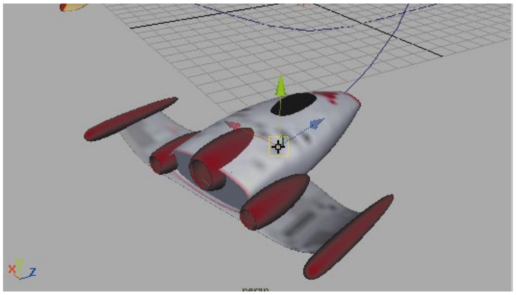
The move manipulator is currently oriented to the world space grid. You need it to move in the local space of the light.

- Double-click on the **Move** tool icon or select **Modify** → **Transformation Tools** → **Move** - ❐ and set the following option:

  Move to **Local**.

- Click **Close**.

  Now the orientation of the move manipulators are aligned with the ship group.

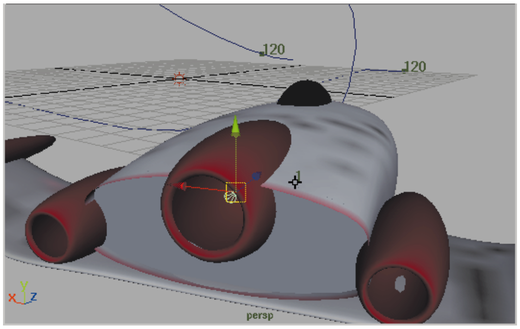


Local move manipulator

3 Position the light

You can now position the light using the local axis to keep yourself aligned with the nurbShip.

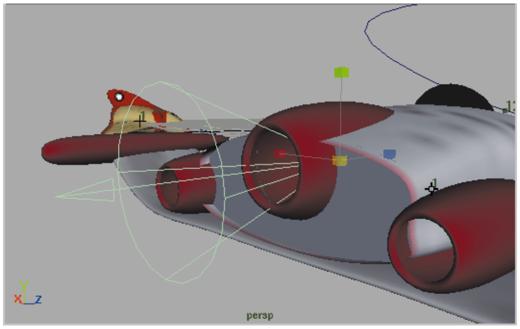
- **Move** the light so that it sits in the center of the thruster.



Positioned light

- **Scale** the size of the light.

  The light cone should now protrude outside of the thruster. This will make it easier to select.



Scaled light icon

4 Change the spot light color

- Open the Attribute Editor and in the **Spot Light Attributes**, set the following:

  Color of the light to a bright orange using values of **RGB** values of about **1.0**, **0.33**, **0.15**.

  Decay Rate to **Cubic**;

  Cone Angle to **60**.

  Penumbra to **2.0**

5 Save your work

- Select **File → Save Scene As** and enter the name *SpaceScene2*.

Adding the fog effects

You will now add fog to the spot light in order to create the thruster flame effect. You will set the fog attributes then test render the effect.

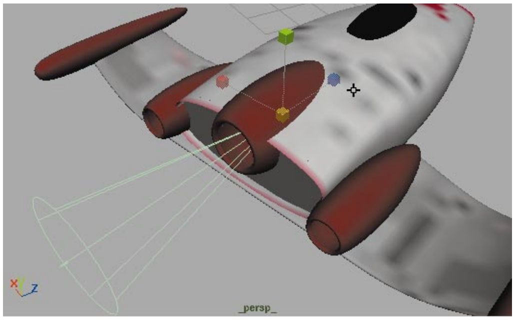
1 Add fog to spot light

- Open up the **Light Effects** section in the Attribute Editor and click on the **Map** button next to the **Light Fog**.

2 Rescale the light

The fog adds a *coneshape* icon to the spot light to show the extent of the fog. You may want to scale the light back down to shorten the fog's *coneshape* icon.

- **Scale** the *coneshape* icon until the icon is closer to the ship.



Fog coneshape icon

3 Adjust the spot light shape node

- In the Attribute Editor, click on the *spotLightShape* tab and in the **Light Effects** section set the following:

  Fog Spread to **0.5**;

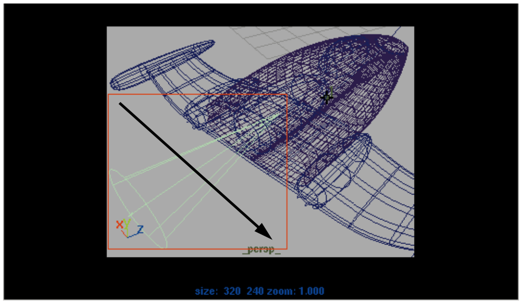
  Fog Intensity to **5.0**.

4 Test render the effect

You will be using the IPR feature in the Render View window to fine tune the look of the thruster fire.

- Select a closeup view of the backside of the ship.
- In the Render View, select **IPR → IPR Render → Persp.**

  This will generate an IPR file.
- **Click-drag** select a region around the thruster for fine tuning



Selected region

You will be adjusting some of the attributes in the Attribute Editor and the IPR render will update instantly.

Animate the fog effect

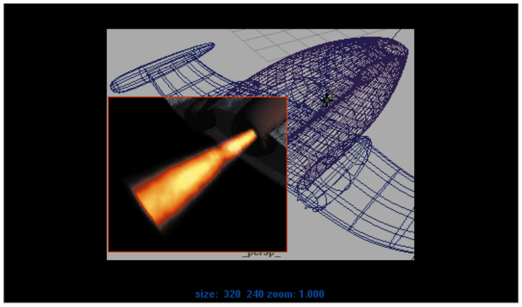
To create a thruster flame that has an animated effect, you will apply a 3D fractal texture. You will then animate the position of the fractal to give the fog some motion.

1 Add a fractal map to the fog

- Click on the *lightFog* tab in the Attribute Editor.
- In the **Light Fog Attributes** section of this new node, click on the **Map** button next to **Color**.
- Select a **Solid Fractal** 3D texture.
- **Hide** the 3D texture icon.

  You should notice that the IPR render has updated to reflect the change in fog.
- Fine tune the thruster fire to your liking.

Note: If you change the cone angle, it is recommended that you regenerate the IPR file.



Rendered region showing fractal map

2 Animate the fog effect

You do not actually have to do anything to animate the exhaust. Because the fractal map's placement icon is fixed at the origin, the look of the fog will change over time as the ship animates along its path.

In essence, the fog light is passing through the fractal effect as the ship animates. This makes sure that the effect is different for different frames.

Building a photon torpedo

You will now animate a point light shooting from the nurbShip to the polyShip.

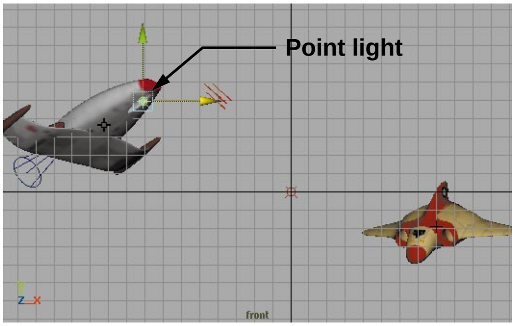
1 Place a point light in scene

- Select **Lights** → **Create Point Light**.
- Turn off the IPR updating by pressing the red button in the top right hand corner of the Render View window.

2 Animating the point light

- Go to frame **30**.
- **Move** the point light up into the photon recess of the nurbShip.

  You may need to use different orthographic views to place the light correctly.
- Press **Shift-w** to set keys on the three translation channels.



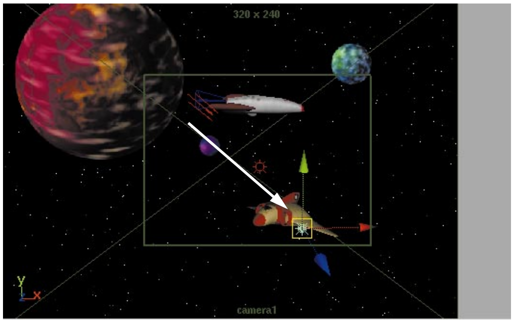
Point light position at frame 30

3 Set a second key for the light

- Go to frame **70**.
- **Move** the point light until it sits just in front of the left wing of the polyShip.

  Again, you will need to use different views to position this where you want it to go. This position is meant to represent a near miss as the torpedo grazes the polyShip's wing.
- Press **Shift-w** to set keys on the three translation channels.
- Play back the results in the *camera1* view.

The point light should remain stationary until frame 30, then it will move to its second position at frame 70.



Point light position at frame 70

4 Animate the light's visibility

You only want to see the light from frame 30 until frame 85.

- Go to frame **1**.
- In the Channel box, change the **Visibility** to **Off**.
- Click on the **Visibility** channel with your right mouse button and select **Key Selected** from the menu.
- Go to frame **30**.
- In the Channel box, change the **Visibility** to **On**.
- Click on the **Visibility** channel with your right mouse button and select **Key Selected** from the menu.
- Play back the results in the *camera1* view.

5 Adjusting the light glow

- Go to frame **55**.
- Create an IPR render and select a region around the light.

- Press **Ctrl-a** to open the Attribute Editor.
- Click on the **Map** button next to **Light Glow**.
- In the *opticalFX* node, change the following attributes:

  Glow Type to **Linear**;

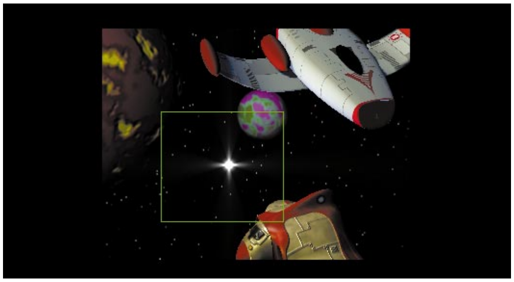
  Radial Frequency to **1.0.**
- In the **Glow Attributes** section:

  Glow Color to a bright blue using RGB values of about **0.0, 0.465, 1.0**;

  Glow Intensity to **1.5**;

  Glow Spread to **0.66**;

  Glow Noise to **0.2**.



IPR Rendered region

Animate the effect

You will now set keys on some of the optical effect attributes belonging to the point light.

1 Animate the light attributes

- Go to frame **30**.
- Click with your right mouse In the Attribute Editor on the following attribute names, one at a time, and select **Set Key**:

  Radial Frequency;

  Glow Spread;

  Glow Noise.
- Go to frame **70** and set another set of keys for the same attributes.
- Go to frame **50**.
- Set the following:

  Radial Frequency to **0.5**;

  Glow Spread to **0.5**;

  Glow Noise to **0.0**.
- Set keys for these attributes.

2 Fade out the photon light

- Go to frame **70**.
- In the Attribute Editor, open the **Optical FX Attributes** section of the *opticalFX* node.
- Set a key for the **Glow Intensity** attribute.
- Go to frame **80**.
- Set the **Glow Intensity** to **0.0**.
- Set a key on this attribute.

  You will see the results of these settings later when you render the scene.

Building an explosion

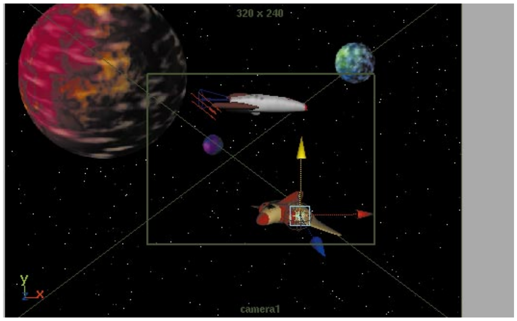
Using a new point light, you will set keys for a number of light attributes to create the effect of an explosion which will occur at the tip of the polyShip's wings.

1 Create a new point light

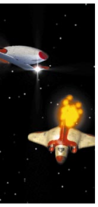
- Select **Lights → Create Point Light**.

2 Animate point light coming into scene

- Go to frame **70**.
- **Move** the light to the same position as the first point light.



New point light



3 Edit the point light attributes

- Create an IPR render and select a region around the light.
- In the Attribute Editor, open the **Point Light Attributes** section and set the following;

  Color to bright orange;

  Decay Rate to **Linear.**

  In the **Light Effects** section:

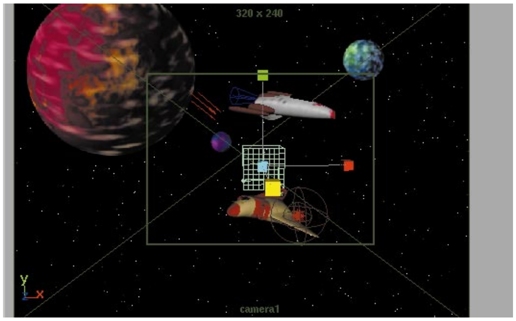
  Fog Type to **Linear;**

  Fog Radius to **2**.

- Click on the **Map** button next to **Light Fog.**
- In the new fog node, click on the **Map** button next to **Color**.
- Select a **Solid Fractal** 3D texture.

4 Scale the fractal map

- In the Perspective view, **select** the green 3D texture icon.
- Press **Shift-r** to set keys for the scale attributes.
- Go to frame **80**.
- **Scale** the icon by **2** units in all three axes.
- Press **Shift r** to set keys for the scale attributes.



Scaled texture icon

- Go to frame **70**.
- Click-drag with your middle mouse button in the Time slider to go to frame **90**.

  The middle mouse button lets you use the attribute values from frame 70 for your new frame.

- Press **Shift-r** to set keys for the scale attributes.

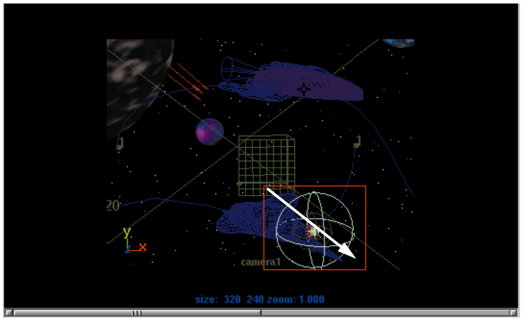
5 Animating the light Intensity

- Select the explosion point light and in the Attribute Editor, click on the *pointLightShape* tab.
- Go to frame **70**.
- In the **Point Light Attributes** section, set the **Intensity** to **0.0** and set a key.
- Go to frame **90**.

- Set another key on the **Intensity** attribute.
- Go to frame **80**.
- Set the **Intensity** to **2** and set a key.

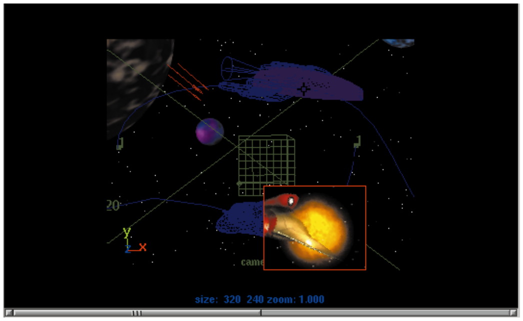
6 Test render the effect

- Go to frame **80**.
- From the Render View panel, select **Render** → **Snapshot** → **camera1**.
- Click-drag a selection box around the new point light.



Selected region

- Select **Render** → **Region**.



Rendered region

7 Adding and animating halo

- In the Attribute Editor's **Light Effects** section, click on the **Map** button next to **Light Glow**.
- In the new *opticalFX* node, set the following:

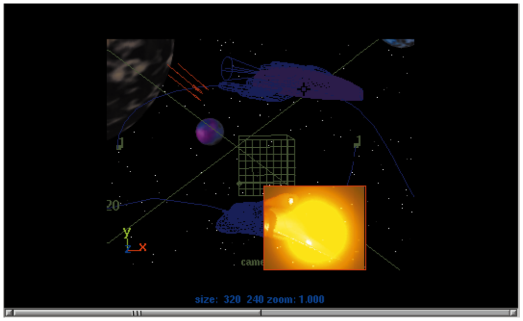
  Glow Type to **None**;

  Halo Type to **Linear**.
- Go to frame **70**.
- Set the **Halo Intensity** to **0**.
- Set a key on this attribute.
- Go to frame **90**.
- Set another key on this attribute.
- Go to frame **80**.
- Set the **Halo Intensity** to **15**.
- Set a key on this attribute.

8 Test render the effect

- Go to frame **80**.
- From the Render View panel, select **Render** → **Region**.

  The effect extends beyond the region but the general look can be easily evaluated.



Rendered region

PROJECT THREE

Updating the polyShip animation

At the moment, the polyShip is flying right past the explosion as if nothing is happening. You will now set keys on the path animation's *twist* attribute to animate the ship reacting to the explosion.

When you set keys on the *motionPath* node's twist attribute, you will create new *orientation markers* that will work similar to the position markers which you learned about in the previous lesson.

1 Select the polyShip's path node

- Select the polyShip using its selection handle.

  This should pick the node's *polyShipAnim* node.

- In the Channel box, click on the *motionPath* input node.

2 Unlock the front twist node

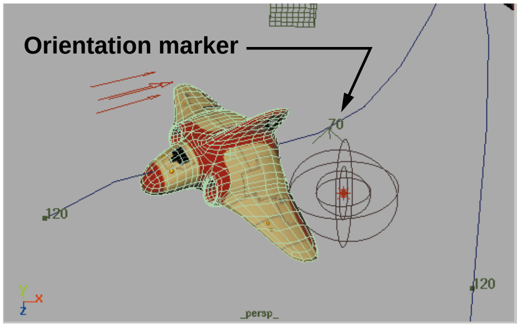
The three twist nodes are currently locked. You can see this because their value fields in the Channel box are highlighted in gray.

- In the Channel box, click on the **Front Twist** node name with the right mouse button and select **Unlock Selected** from the pop-up menu.

3 Set a key on the twist

- Go to frame **70**.
- In the Channel box, click on the **Front Twist** attribute name with the right mouse button and select **Key Selected** from the pop-up menu.
- Go to frame **90**.
- Set another key.

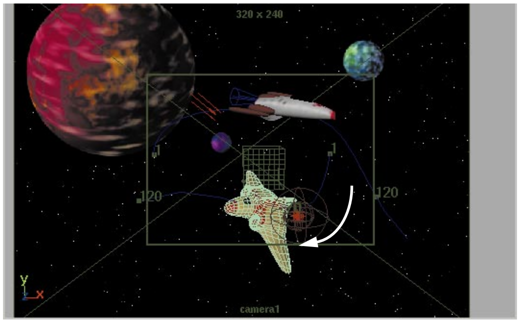
These keys place orientation markers onto the path. These markers can be edited using similar methods used on the position markers you learned about in the last lesson.



Orientation marker

4 Set a key on the twist at frame 80

- Go to frame **80**.
- Highlight the **Front Twist** attribute in the Channel Box.
- In the *camera1* view, **click-drag** with the **MMB** to twist the polyShip about -40 degrees.
- Set a key on the **Front Twist** attribute.



Twisted ship

- Playback the results

PARTICLE EFFECTS

You will now add software particles to the polyShip. You will first create thruster plasma for the second ship and then you will add smoke to the wing after it is hit by the explosion.

Create a particle emitter

You will start with a particle emitter that will be used for the polyShip's thruster. You need to create the emitter and then parent it to the polyShip just as you did with the light and the nurbShip.

1 Set up your Perspective view

- Set up one of your panels in Perspective view.
- Select **Show** → **None** then **Show** → **Meshes**, **Show** → **Handles**, **Show** → **Dynamics** and **Show** → **Curves**.

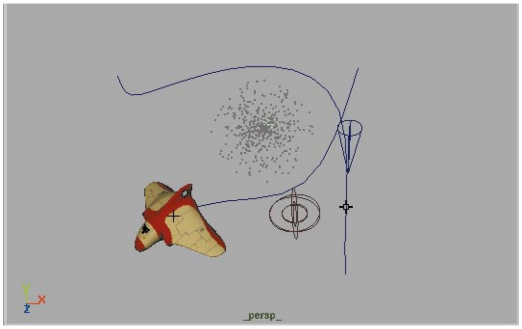
  Now you can preview only the elements needed for this section of the lesson.
- Move your cursor into the Perspective view panel and press the spacebar to pop it to full screen.

  This will make things run a little faster.

2 Create an emitter

- Go to the **Dynamics** menu set.
- Select **Particles** → **Create Emitter.**
- Play back the animation.

  The particles are emitting, but are not moving with the ship. You need to parent the emitter into the ship.



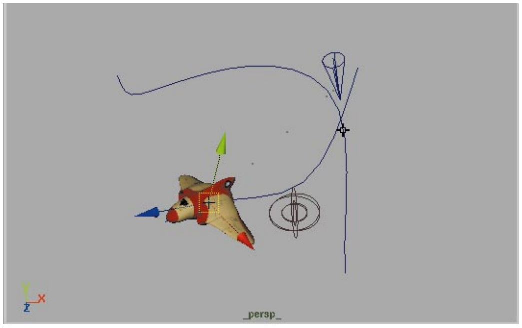
Unparented emitter

3 Parent the emitter to the polyShip

You will now parent the emitter to the ship in order to get them moving together.

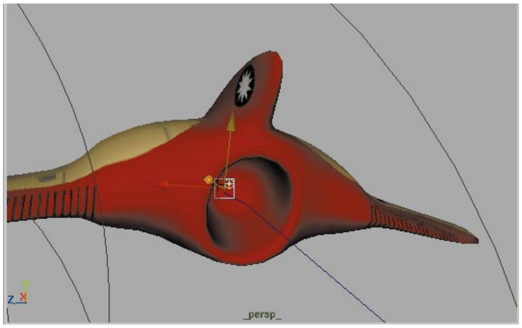
- Press the **Shift** key and select the *polyShipAnim* node using its selection handle.
- Select **Edit** → **Parent.**

  The **Preserve Position** option should still be set to **Off** and the emitter will move to the center of the node.
- Select the **Move** tool. The **Move** option should still be set to **Local**.



Parented emitter

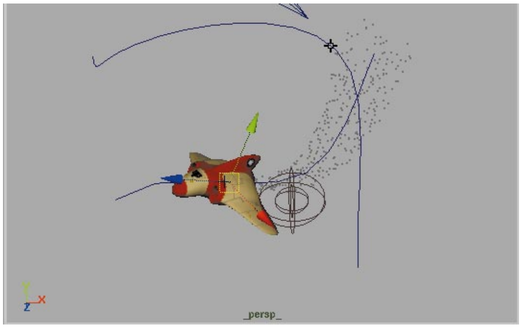
- **Move** the emitter along the local Z-axis to move it into the polyShip's thruster space.



Positioned emitter

- Play back the scene.

  Now the particles are being emitted out of the thruster space.



Particles being emitted along path

4 Set the emitter direction

- Go to frame **1**.
- Make sure that the emitter is selected.
- In the Channel box, change the following attributes:

  Direction X to **0**;

  Direction Y to **0**;

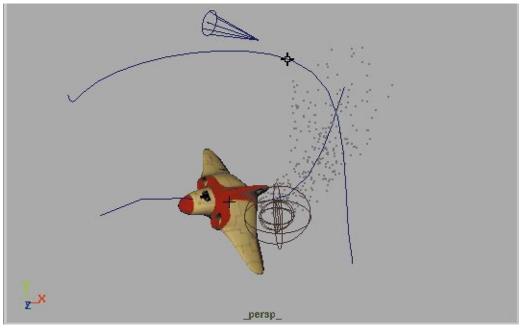
  Direction Z to **1**;

  Spread to **1**;

  Speed to **5**;

  Emitter Type to **Directional**;

- Play back the scene.



Updated particle emission

Particle attributes

Now that you have particles being emitted in the desired direction, you can start to look at the particle node attributes. These attributes would include *lifespan* and *color*.

1 Set particle attributes

- Select the *particle* node by selecting any of the emitted particles.
- Open the Attribute Editor.
- Click on the *particleShape* tab.

2 Add a lifespan attribute

Right now, the particles are remaining on the screen indefinitely. You can use a lifespan attribute to limit how long the particles live.

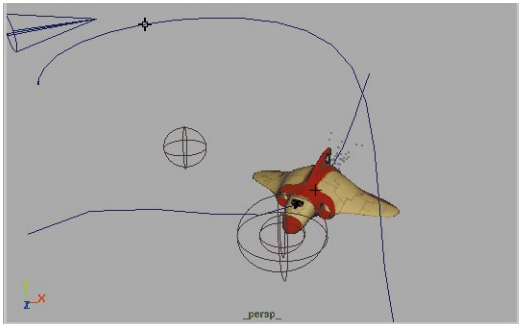
- In the Attribute Editor, open the **Add Dynamic Attributes** section and click on the **Lifespan** button.

  A **Particle Lifespan** window will be displayed.
- Click on **Add Per Object Attribute**.
- Click on **Add Attribute**.

  Notice that a new attribute called **Lifespan** has been added to the Render attributes of the particle node.

3 Edit the lifespan

- Set the **Lifespan** to **0.5**;
- Play back the results.



Particles with shorter lifespan

Rendering the particles

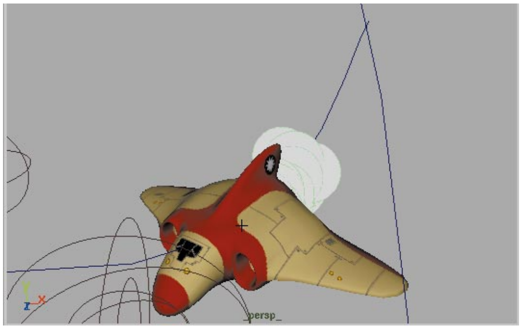
You now want to set up the particles for rendering. In Lesson 4, you used hardware rendered particles to add a spark effect to the bouncing ball and software rendered particles for the ring of fire.

For this lesson, you will again use software rendered particles. These will give you particles which will render with your other objects during the batch rendering process.

1 Change the render type

- With the *particleShape* tab active in the Attribute Editor, go to the **Render Attributes** section and set **Particle Render Type** to **Cloud (s/w)**.

  The *(s/w)* label means that this render type will render with software rendering. This offers a different kind of effect than hardware rendered particles.
- Play back the animation to about frame 55 and stop to see the particle display.



Cloud render type display

2 Add render attributes

Render attributes can be added to the node to help you set up the look of the particles.

- In the **Render Attributes** section, next to **Add Attributes for**, click on the **Current Render Type** button.

This adds three new attributes to the node.

- Set the following:

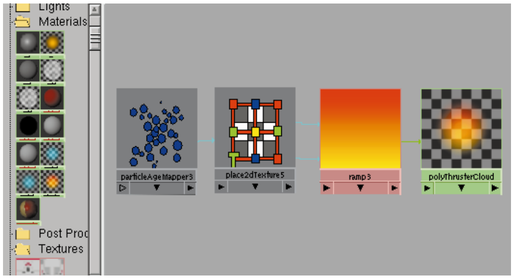
  Radius to **0.5**;

3 Map the particle color

You now want to map the particle with a particle shading group. This will let you define the rendered look of the particles.

- In the Hypershade, select **Create → Materials → Volumetric → Particle Cloud.**
- Click in the Hypershade workspace.
- Rename the Material node *polyThrusterCloud.*
- Open the Attribute Editor for *polyThrusterCloud.*
- Press the **Map** button next to **Life Color**.

  By using the **Life Color** instead of **Color**, it will be connected directly with the lifespan attribute.
- Click on **Ramp**.
- Set **Relative Age** to **On** in the *ParticleAgeMapper.*
- In the Hypershade highlight the ramp node to see it in the Attribute Editor.
- Adjust the ramp to be from yellow at the bottom to red at the top.



polyThrusterCloud in the Hypershade

4 Make the particles fade out

You will now add transparency to the particles. As they get older, they will become more transparent.

- Open the Attribute Editor for *polyThrusterCloud.*
- Press the **Map** button next to **Life Transparency**.
- Click on **Ramp**.
- Set **Relative Age** to **On** in the *ParticleAgeMapper.*
- In the Hypershade highlight the new ramp node to see it in the Attribute Editor.
- Adjust the ramp to be from black at the bottom to white at the top.

5 Assign the shading group

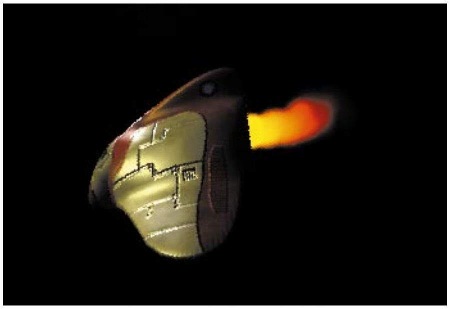
- Highlight the *polyThrusterCloud* node in the Hypershade.
- **Select** the particles in the view panels.
- In the Hypershade, with **RMB** over the *polyThrusterCloud* node, select **Assign Material To Selection**.

  The particles turn green to indicate that a shading group has been applied.

6 Test render the particles

- Use **Render → Region** to test render the particles coming out of the ship.

  Once you finish, you can return and tweak the various attributes to get the look you want, then test render.



Rendered particles

Particle smoke

The next effect is going to be particle smoke that will be emitted from the polyShip's wing after being grazed by the explosion. This will be created in a similar manner to the last section, except that now you must set the particles to start emitting after frame 80.

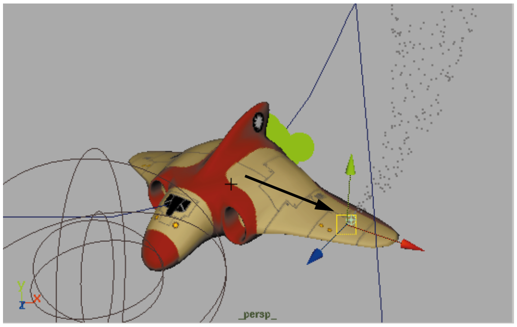
1 Create the emitter

- Select **Particles → Create Emitter.**

2 Parent the emitter

Again the emitter must be parented to the ship in order to move with it.

- **Parent** the emitter to the polyShip.
- **Move** the emitter in local space to just above the wing of the spaceship as shown in the following:



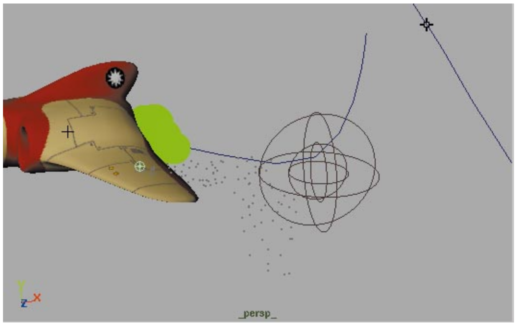
Parented emitter

3 Set keys on the emission

You will now set keys on the rate of emission so that the emitter starts with no particles, then starts up after frame 80.

- Go to frame **80**.
- In the Channel box, click on the *emitterShape* input node and set the **Rate** to **0**.
- Use your right mouse button to select **Key Selected** for the **Rate**.
- Go to frame **85**.
- Set the **Rate** to **100**.
- Set another key.

Now the particles will begin emitting right after the explosion.



Particles after frame 85

PROJECT THREE

4 Set the particle attributes

You will again use a cloud Render Type. The steps will be almost identical to the thruster particle set-up.

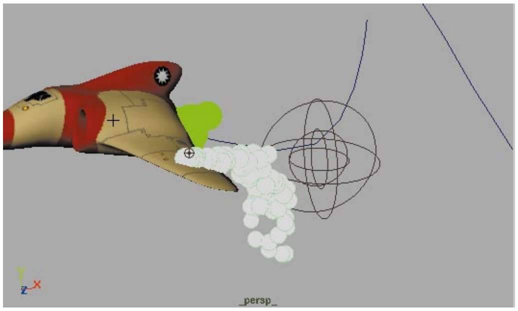
- Select the particles and open the Attribute Editor.
- Add lifespan to the node and set it as follows:

  Lifespan to **0.75**.

- Set the Render Type to **Cloud(s/w)** then add the render attributes and set them as follows:

  Radius to **0.25**;

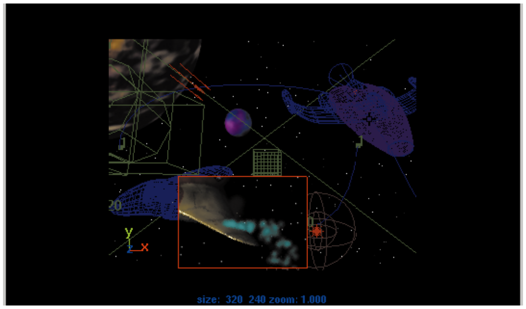
  Threshold to **0.75**.



Particle render type

5 Test render the particles

- Use **Render → Region** to test the particles at around frame 90.



Rendered particles

RENDERING

You are now going to batch render the scene to preview the results. You are going to create a very low quality test of this animation, in order to test all of the cool effects you have integrated into the scene.

It is always a good idea to preview a low resolution/low quality test before committing to a longer rendering. While still image tests may offer you some idea of how the scene may appear, it is always best to see if the sequence animated properly.

Setting the Render Globals

You have already used the Render Globals window to set up some test renderings. This window is where you make your final decisions before sending a rendered animation off to the batch renderer.

1 Open the Render Globals window

- Select **Window → Render Globals...**

2 Set the Image File Outputs

- Set **Frame/Animation Ext** to **name.ext.#.**

- Set the **Start Frame** to **1**, and the **End Frame** to **120**.
- Turn **Alpha Channel (Mask)** to **off**.

3 Set the Anti-Alias Quality

- Open the **Anti-Alias Quality** Section.
- Set the **Presets** to **Preview Quality**.

4 Set the size

- Open the **Resolution** Section.
- Set the **Render Resolution** to **320x240**.

5 Save your work

- Select **File** → **Save Scene As**. Enter the name *spaceSceneComplete* and press **Save**.

6 Batch render the scene

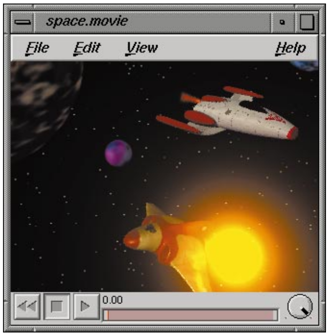
- Select **Render** → **(Save) Batch Render**.
- Enter the name *spaceSceneTest1* and click on the **Batch Render** button.

  Refer to the Feedback line in the lower right to follow the progress of the rendering.

7 Previewing the animation

Once you have completed the animation, you can view your work as a Playblast animation or by using *fcheck*.

If you are happy with the results, then you can do another test at a slightly higher quality.



Movie Player preview

The final rendering

While there is no need to create a final high quality rendering of this lesson, it is worthwhile to think about some options:

1 Rendering the whole scene

The first option is to create a finished animation of the whole scene with all of the parts working together. This seems at first to be the easiest solution because it means you would render just once to get the final result.

2 Rendering the scene in layers

The other option is to render each part of the scene separately. These different render *passes* can later be composited. For this lesson, you might render the following layers:

- background space scene
- nurbShip
- polyShip
- photon torpedo OptiF/X
- explosion OptiF/X
- nurbShip thruster OptiF/X
- smoke particles

PROJECT THREE

- polyShip thruster particles

These passes put each key object onto their own layer when you composite. You also avoid rendering the scene with an image plane background which is more easily accomplished in a compositing package.

Now, if changes are to be made to one of the elements – say, a texture map to be modified for one of the ships, or one of the glows to be intensified – then only that layer has to be updated. Also, color corrections can be made to each layer if desired.

Should you want to render these passes separately, then you will need to hide the other objects and render with the Mask option turned on in the Render Globals. To render the OptiF/X and particle layers, you may need to apply a special **Use Background** material so that the mask is more accurate. Refer to the *Using Maya: Rendering* guide to find out more about this shading group type.

Resetting the move tool

Earlier you changed the Move tool options to Local. You should change it back to Global to make sure that upcoming lessons work properly.

1 Change the Move tool options to world

- Double-click on the **Move** tool icon or select **Modify** → **Transformation Tools** → **Move -** ❐ and set the following option:

  Move to **World**.

Conclusion

You have now touched on the visual effects offered by Maya’s OptiF/X features such as fog and glow, while comparing these to Maya F/X’s particle effects.

In the next lesson, you will create a walking biped character called Primitive Man. You will then apply character deformations to make Primitive Man’s elbows tuck and his arms bulge.

Lesson 16

Primitive Man

Lesson 17

Animating a Walk Cycle

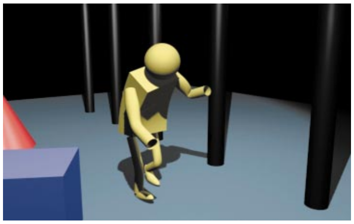
Project Four

Project Four

In this project, you are going to model, texture and animate a walking character – Primitive Man. This character will be built using skeleton joints that are controlled using IK solver handles. Expressions and constraints will then be used to create a rolling foot motion.

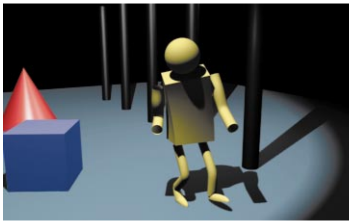
The surfaces of this character will be built using primitive shapes. These surfaces will then be bound into the skeleton. Character deformers known as *flexors* will then be added to control the bending of the knee and elbow, and the bulging of the upper arm.

1. The camera follows as Primitive Man walks in his primitive world.



2. As he explores his surroundings, he bends down to check his footing.

3. He is slightly distracted by something happening high to his left.



4. Satisfied, Primitive Man strolls forward with confidence.

STORYBOARD

16 Primitive Man

In this lesson, you will build a character out of skeleton joints and primitive shapes. The character's motion will be established using IK chains on the arms and legs while expressions will help you control the roll of the foot.

Primitive Man

You will then bind the primitives into the skeleton in order to get skeletal deformations. In cases where the surface is not suited for being bound directly, you will apply a *lattice* and bind the lattice instead. You will then use joint and bone *flexors* to further refine the surface deformations.

In this lesson you will learn the following:

- How to create skeleton joints with IK handles
- How to create a rolling foot hierarchy
- How to bind surfaces and lattices into skeleton joints
- How to edit the membership of a cluster set
- How to use flexors to tuck and bulge a surface

Initial set-up

You will now create a new project then create the files from scratch.

1 Create a new project

- Select **File** → **Project** → **Set...**
- From the *learningMaya* directory, set the name to *projectFour*.
- Click **OK**.

2 Set up a four view panel layout

- Select **File** → **New...**
- Set up a four view panel layout that shows the three orthographic views and a Perspective view.

BUILDING A CHARACTER

Building a character usually involves building the character's surfaces, drawing skeleton joints to match the surfaces. The surfaces are then bound to the skeleton and deformations are applied.

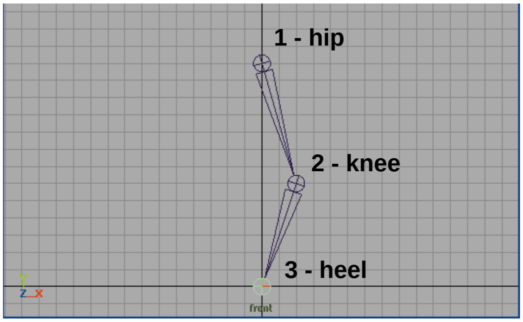
For this lesson, you will build the skeleton first, then build surfaces to suit. This will let you explore skeletal issues before worrying about the surfaces.

Drawing a skeleton leg

You will start by creating the skeleton's legs. This will involve creating the joints and later you will add IK chains.

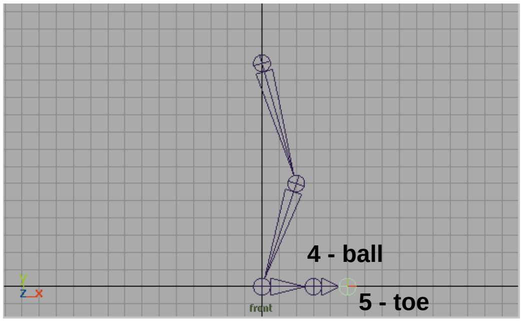
1 Draw the leg joints

- Set your Grid to default settings by selecting **Display** → **Grid**.
- Go to the **Animation** menu set.
- In the Front view, track the view to place the X-axis line at the bottom of the view.
- Select **Skeletons** → **Joint Tool**.
- In the Front view, press the **x** key to use grid snapping, then click three times to place the following leg joints:



Skeleton leg joints

- Now click two more times using grid snapping to place the following foot joints:



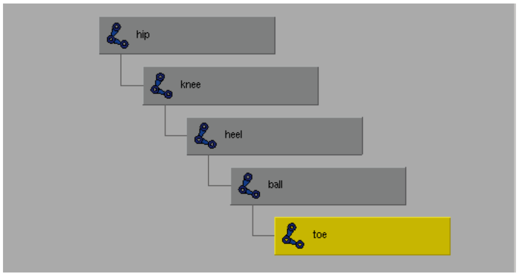
Skeleton foot joints

- Press the **Enter** key to accept the joints.

2 Rename the joints

- Open a Hypergraph view panel.

- Dolly and track to see the new joints.
- Rename the joints as shown below. Use the names as labelled in the last two diagrams:



Renamed joints

Note: In this skeleton, the ankle joint has been left out. To simplify this example, you will rotate the foot around the heel instead.

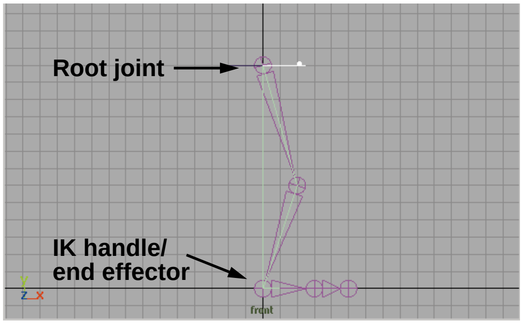
Adding IK chains to the leg

To control the leg, you will need two IK chains – one running from the hip to the heel and another from the heel to the ball of the foot. These chains will be used to control the motion of the leg.

1 Draw the initial IK chain

- Select **Skeletons → IK Handle Tool**.
- Click on the *hip* joint.
- Click on the *heel* joint to create an IK chain.

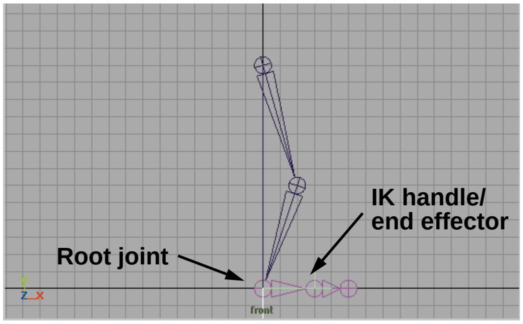
  This chain uses a *rotate plane* IK solver which offers more control than the single chain solver.



First IK chain

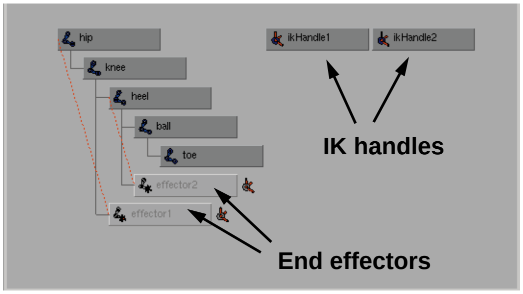
2 Draw the second IK chain

- Press the **y** key to repeat the last tool.
- Click on the *heel* joint again.
- Click on the *ball* joint to create a second IK chain.



Second IK chain

In the Hypergraph, you can see the end effectors connected into the hierarchy and the IK handles to the side. The end effectors and the IK handles are connected along with the appropriate joints at the dependency node level. When you control the handles, you control the whole IK chain.



Nodes in Hypergraph

3 Group the IK handles

Since you generally want these two IK handles to work together, you should group them and use the group to control the IK chain movement and rotation.

- **Select** the two lK handles.
- Select **Edit** → **Group**.
- In the Channel box, change the name of the group to *rfootc*.

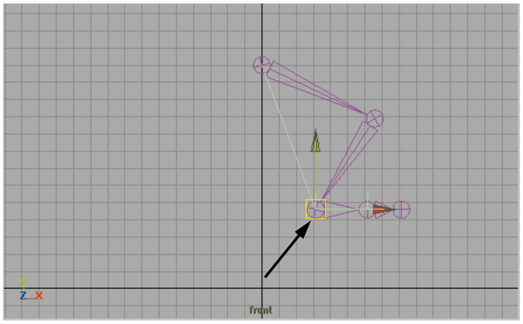
  This is meant to represent "right foot control" but has been shortened to make it easier when you write expressions later.
- Select **Display** → **Object Components** → **Selection Handles**.

  You will be able to use this handle easily to select the group later.

4 Test the movement of the group

- **Move** the control group to move the foot.

  You can see that the foot moves while the hip stays in places.
- Press **z** to undo this move.



Moved group

Rolling the foot

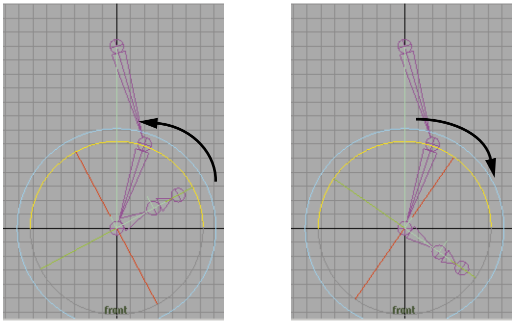
Right now, this group can be used to control the movement of the foot, and also to control the *rolling* of the foot. This way, you are able to direct what happens as the heel of the foot touches the ground, and as it rolls onto the ball of the foot.

You should note that this foot assembly will be built to only handle the forward motion of primitive man. If you wanted to have him turn a corner, then you would have to build an assembly that takes into account rotations in all directions.

1 Rotate the group

- **Rotate** the group around the Z-axis using the rotate manipulator.

  When you rotate in foot backwards, the foot raises as a group which is correct. When you rotate in the forward direction, the foot rotates down below the ground which is not what you want.



Forward rotation **Backward rotation**

Rotated group

- When you are finished, press **z** to undo until your foot is back to the starting position.

2 Create an expression for the group

You are now going to write an expression that sets the location of the group's pivot point based on its rotation around Z.

- Select **Window → Expression Editor...**
- In this window, set the **Expression name** to *roll1*, then in the **Expression:** field enter the following expression:

```
//expression to control
//roll of the right foot

if ( rfootc.rz >= 0 )
   rfootc.rotatePivotX = 0;
else
   rfootc.rotatePivotX = 3;
```

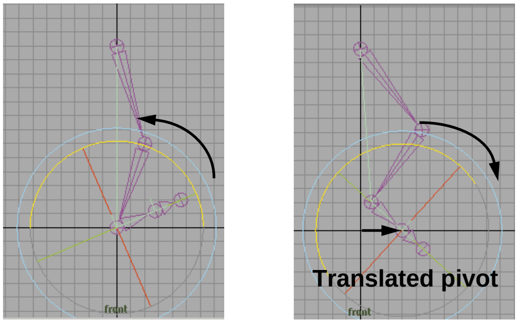
- Click on the **Create** button.

3 Rotate the group

You can again rotate the group to see the effect on the feet joints.

- **Rotate** the group around the Z-axis using the rotate manipulator.

  When you rotate in the backward direction, the foot raises as a group just like the last time around. When you rotate in the forward direction, the foot rotates around the ball of the foot, since the pivot has been translated by the expression. This effect is getting closer to what you want.



Positive rotation **Negative rotation**

Rotated group with expression

- When you are finished, press **z** to undo until your foot is back to the starting position.

Orienting the toe

You now want the toe to stay straight when the foot rotates back, then stay aligned with the ground when the foot rotates forward. You will accomplish this with an orientation constraint and an expression.

1 Create a null node

- Select **Edit → Create Empty Group**.

  It is called *null1* and is placed at the origin.

- Rename it *ballNull*.

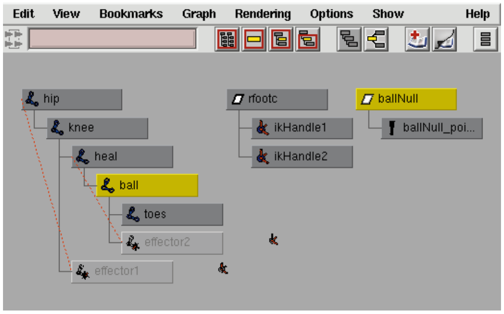
2 Point constrain the null node

- In the Hypergraph, **select** the *ball* joint node on its own.
- Press the **Shift** key and click on the *ballNull* node.

  You need to pick this last if you want to constrain it.

- Go to the **Animation** menu set.
- Select **Constrain → Point**.

  Now the null node will move with the ball of the foot.



Selected nodes

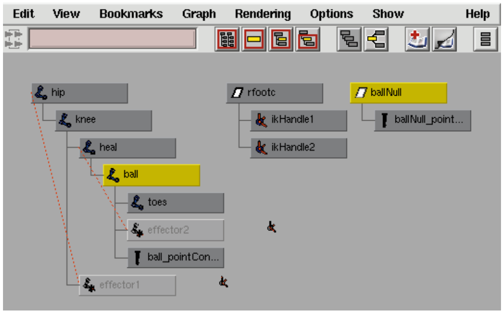
3 Orient constrain the foot's ball joint

- In the Hypergraph, reselect the *ballNull* joint node on its own.
- Press the **Shift** key and click on the *ball* joint node.

  This time you picked the *ball* joint last since it is going to be constrained.

- Go to the **Animation** menu set.
- Select **Constraints → Orient**.

Now the toe will be aligned with the null node which is, by default, aligned with the world space axis.

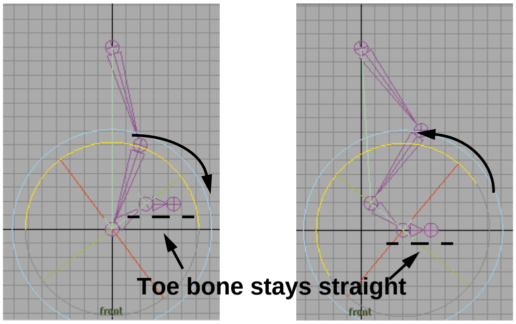


Selected nodes

4 Rotate the group

- **Select** the *rfootc* group.
- **Rotate** the group around the Z-axis using the rotate manipulator.

  When you rotate in the positive direction, the foot raises as a group – except now the toe is pointed forward, which you don't want. What will work is when you rotate in the negative direction, so that the foot rotates around the ball of the foot and the toe stays straight.



Backward rotation **Forward rotation**

Rotated group with constraints

When you are finished, press **z** to undo until your foot is back to the starting position.

Note: With this set-up, the knee may flip if you rotate the foot too far. This flipping can be corrected by working with the Rotate Plane IK solver's Pole Vector. For now, just make sure you don't rotate the group far enough to flip.

5 Add another expression

You will be creating an expression to turn the Orient Constraint off when the heel is rotated backwards.

- Select **Window → Expression Editor...**
- In this window, set the **Expression name** to *toe1*, then in the **Expression:** field enter the following expression:

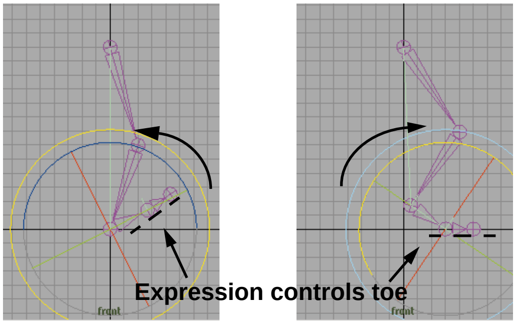
```
//expression to control
//orientation of the right toe
if ( rfootc.rz <= 0 )
ball_orientConstraint1.nodeState
= 0;
else
ball_orientConstraint1.nodeState
= 2;
```

- Click on the **Create** button.

6 Rotate the group

- **Select** the *rfootc* group.
- **Rotate** the group around the Z-axis using the rotate manipulator.

When you rotate in the positive direction, the foot raises as a group — except now the toe is pointed forward, which you *do* want. When you rotate in the negative direction, the foot rotates around the ball of the foot and the toe stays straight – which is correct.



Backward rotation **Forward rotation**

Rotated group with new expression

- When you are finished, press **z** to undo until your foot is back to the starting position.

Creating geometry

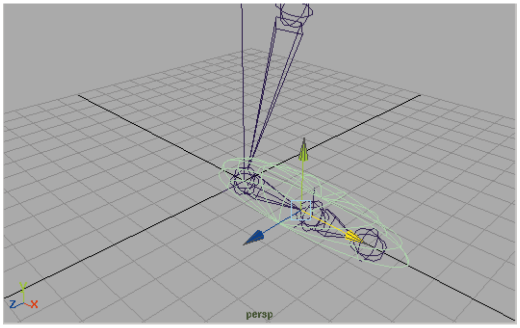
The purpose of creating skeleton joints is to drive the movement of geometry that you can either group or bind to the joints. You will now create a half-sphere to use as the foot and a cylinder to use as the leg.

1 Create the foot using a half-sphere

- Go to the **Modeling** menu set.
- Select **Create** → **NURBS Primitives** → **Sphere**.
- In the Channel box, name this surface *rightFoot*.
- In the Channel box, set the following attributes for the sphere's transform and input nodes:

  Rotate X to **-90**;

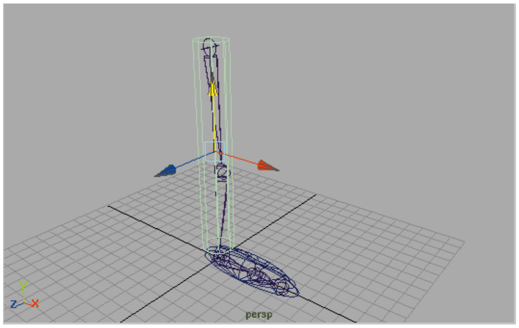
  End Sweep to **180**.
- **Scale** and **Move** the sphere to until it covers the foot joints of the skeleton.



Half-sphere placed and scaled

2 Create the leg using a cylinder

- **Select** the skeleton's *hip* joint node.
- **Move** it up to straighten the leg.
- Select **Create** → **NURBS Primitives** → **Cylinder**.
- In the Channel box, rename this surface to *rightLeg*.
- **Scale** and **Move** the cylinder to point where it covers the leg joints of the skeleton.

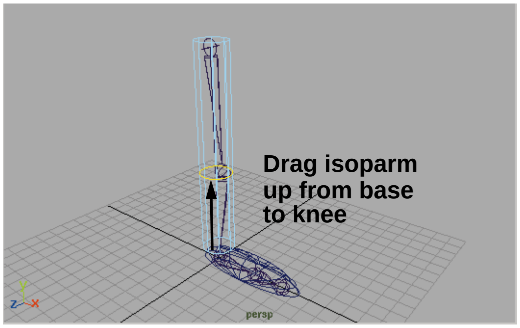


Cylinder covering leg joints

3 Select isoparms on the cylinder

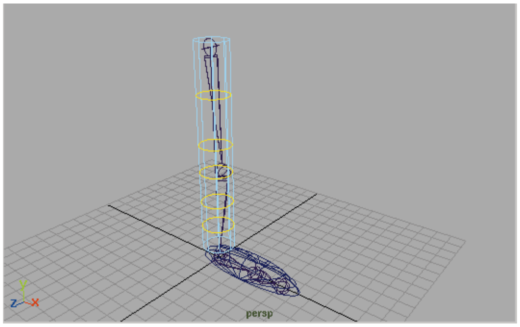
This cylinder is not complex enough to bend properly when you bind it later. You are now going to insert isoparms onto the surface to make it complex enough to deform.

- Click on the cylinder with your **MMB** and select **Isoparm** from the marking menu.
- **Click-drag** on the bottom round isoparm and drag it up to about the area of the knee.



Cross section isoparm selected

- Press the **Shift** key and add more isoparms to the selection by again click-dragging on the base isoparm.
- Use the **Shift** key to select two isoparms around the *knee* joint, and then two others separating the thigh and calf areas.



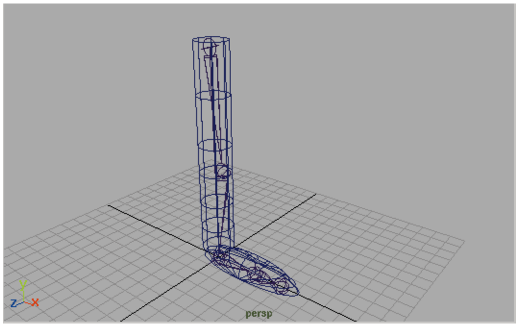
Selected isoparms

4 Insert isoparms on the cylinder

The selected isoparms can now be inserted onto the surface. These lines will be used to subdivide the existing cylinder.

- Select **Edit Surfaces → Insert Isoparms**.

Now you have 6 spans along the length of the cylinder. This will help with the deformation of the cylinder later.



Inserted isoparms

5 Delete history

You are now going to delete history on the two primitives. This will remove the input nodes before you bind the surface.

- **Select** the half-sphere and the cylinder.
- Select **Edit → Delete by Type → History**.

Binding the geometry

To get the surfaces to deform properly, you must bind them to the skeletons. In the case of the sphere, you will apply a lattice and use that object to bind to the joints. You can bind many different object types to skeletons.

1 Add a lattice to the foot

Since the half-sphere has isoparms that meet at a point in the middle of the foot, you may not get the deformations you require. Rather than rebuilding the sphere, you can add a lattice that can be more cleanly bound to the joints.

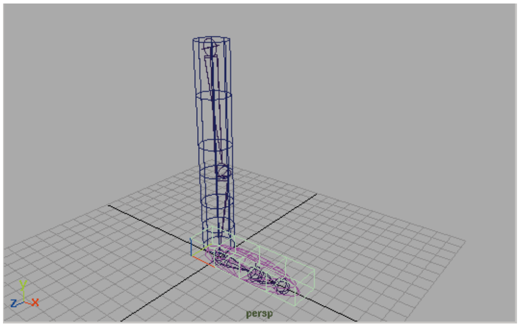
- **Select** the half-sphere surface.
- Go to the **Animation** menu set.
- Select **Deform → Create Lattice**.
- In the Channel box, set the following shape node attributes:

  S Divisions to **5**;

  T Divisions to **2**;

  U Divisions to **2**.

  The lattice is now oriented along the length of the foot. The lattice points will now bind nicely into the skeleton.



Lattice applied to half-sphere

2 Bind the cylinder and the lattice

- **Select** the lattice and the cylinder.
- Press the **Shift** key and select the leg skeleton's *hip* joint node.
- Select **Skinning → Bind Skin - ❐** and set the following option:

  Color Joint to **On**.
- Click **Bind** then **Close**.

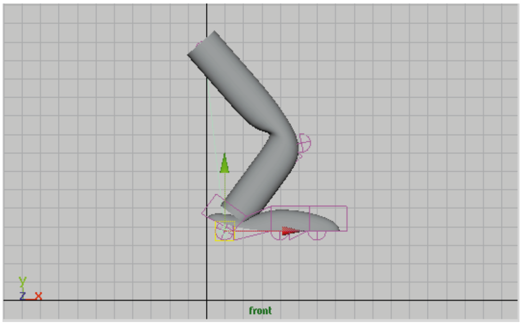
  The joints and bones are now colored differently. This helps you later when you check the membership of CV and lattice points to the bind sets.

3 Test the results

- Activate the Front pane and press **5** to smooth shade that view.
- **Select** the two surfaces and press **3** to increase the surface smoothness.
- **Select** the *rfootc* node using its selection handle.
- **Move** the group to see the effect on the surfaces.

  The foot surface is bending at the back with the leg. Also, the inside of the knee is pinching.

You can fix both of these deformation effects to refine the results of the bind.



Deforming surfaces

Editing the sets

When you bound the lattice and the cylinder to the joints, sets were created. One set is created for every bound joint. As you move and rotate joints, the sets react accordingly.

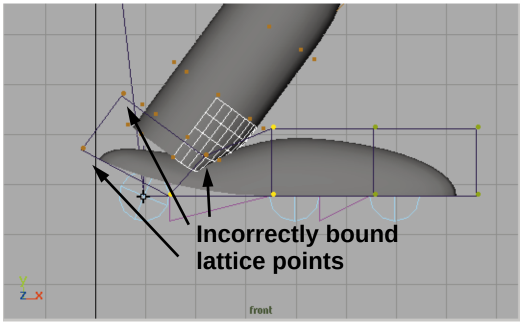
In the case of the back of the foot surface, some of the lattice points are associating themselves with the *knee* joint instead of the *heel* joint. You will now move these points into the *heel* joint set to get the expected results.

1 Edit the set membership on the foot

With the knee in the bent position, select the *heel* joint.

- Select **Deform → Edit Membership tool**.

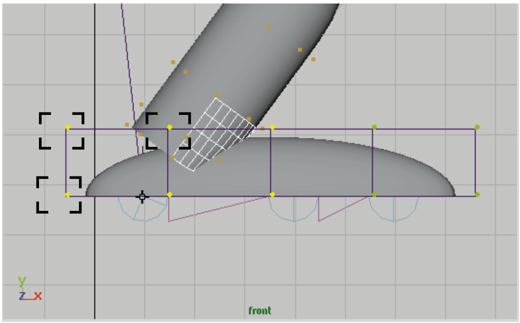
The lattice points associated with the heel are highlighted in yellow. Other points on the cylinder and the lattice are displayed with other colors that match the coloring placed on the skeleton by the bind tool.



Highlighted CV and lattice points

- Press the **Shift** key and with the left mouse button, **Select** the lattice points that make up the heel of the foot.

Once you select the points using this method, they become part of the *heel* joint set and are removed from the *knee* joint set.



Edited membership of points

2 Test the results

- **Select** the *rfootc* node using its selection handle.
- **Move** the group to see how the foot reacts now.
- When you are finished, **Move** the group back to the origin.

Adding a flexor

The other problem you encountered with the deformations was with the back of the knee. You can fix this using a lattice flexor.

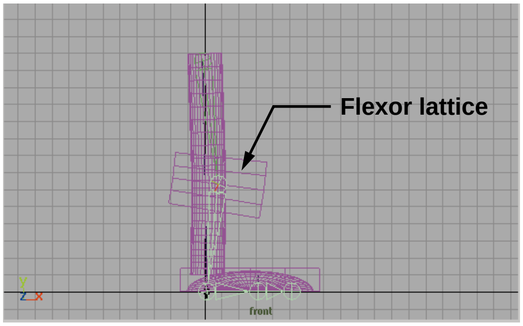
A flexor is a deformer that is automatically bound into a joint hierarchy while being applied to the associated surface.

1 Add a flexor to the knee

- **Select** the *knee* joint.
- Select **Skin → Edit Rigid Skin → Create Flexor...**
- In the Create Flexor window, set the following:

  Flexor Type to **lattice;**

  Joints to **At Selected Joints.**
- Click **Create** then **Close**.

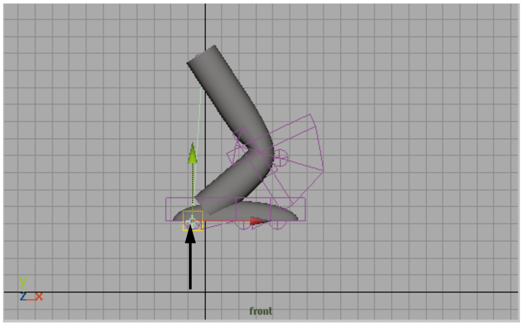


Flexor on knee

2 Test the motion

- **Select** the *rfootc* node using its selection handle.
- **Move** the group to see how the foot reacts now.

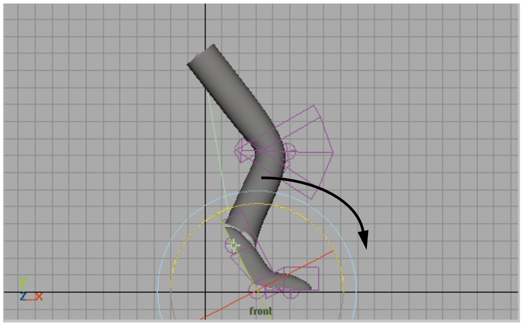
  You are getting a more subtle effect on the inner side of the knee while the outer part of the knee becomes more rounded.



Flexor with moved foot

- **Rotate** the group to see how the foot reacts now.

  You can now see how the foot lattice is bending the foot to align with the toe bone.



Flexor with rotated foot

- **Move** and **Rotate** the group to the starting position.

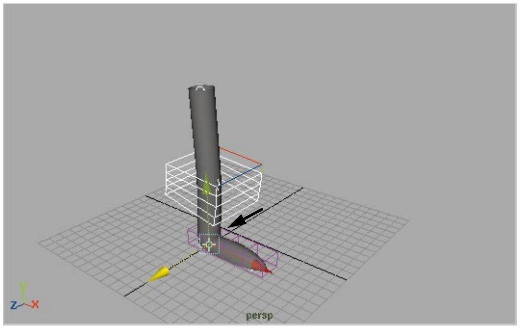
Creating the second leg

You will now create a second leg by duplicating all of the existing hierarchies and their connections. This will give you a second deforming leg.

1 Move the first leg

You will move the *hip* joint and the foot group to one side of the XY plane.

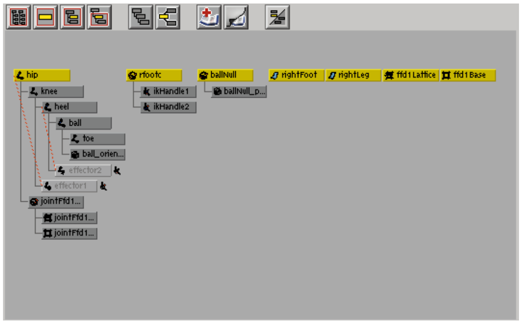
- **Select** the *hip* joint of the skeleton.
- Press the **Shift** key and **select** the *rfootc* node using its selection handle.
- **Move** these two to the character's right side by about 3 units.



Leg moved to the right side

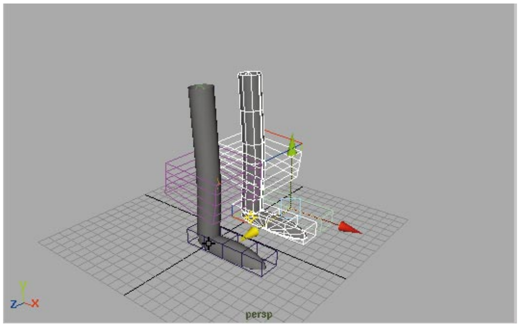
2 Duplicate the first leg

- In the Hypergraph, **select** all of the pieces built so far.



Selected hierarchies

- Select **Edit → Duplicate - ❐** and set **Duplicate Upstream Graph** to **On**.
- Click **Duplicate** then **Close**.
- **Move** the new group by selecting the duplicated *hip1* and *rfootc1* to the character's left side.

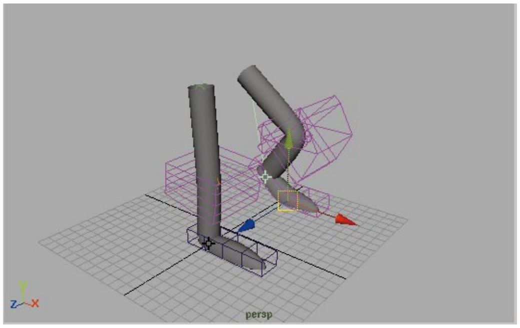


Second leg moved to the left

Note: DO NOT rename any of the new nodes. Renaming could break one of the duplicated expressions.

3 Test the motion on the left leg

- **Select** the *rfootc1* node using its selection handle.
- **Move** and **Rotate** the group to see how the foot reacts now.



Second leg tested

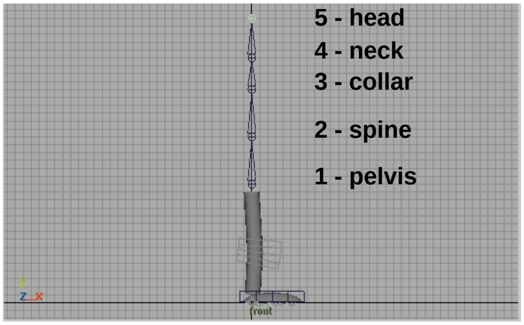
- **Move** and **Rotate** the group to the starting position.

Create the torso and head

To create the character's head and torso, you will start with a hierarchy of joints that will be used to attach a cube for the body and a sphere for the head. In this case, you will not use any IK chains to control the character. Joint rotations will offer sufficient control for the character's spine.

1 Draw the spine joints

- Select **Skeletons → Joint Tool**.
- In the Front view, press the **x** key to use grid snapping, then place 5 joints as shown below. When you finish, rename the joints as labelled:

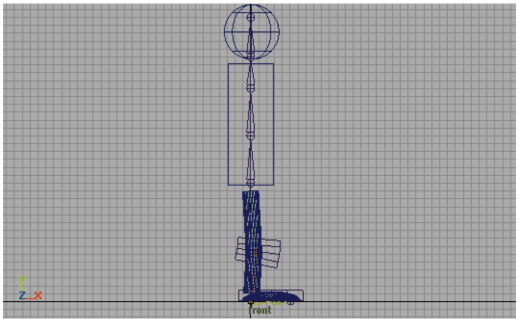


Spine joints

- Press the **Enter** key to finish.

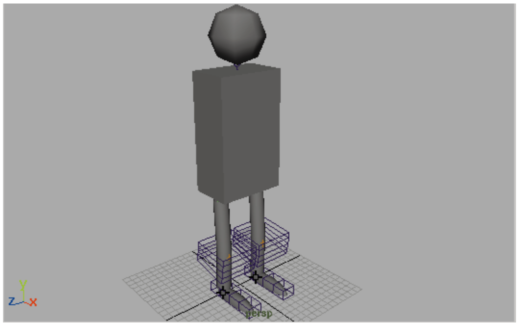
2 Create a head and a body

- Create a Nurbs primitive sphere for the character's head and a Nurbs primitive **Cube** for the body.
- **Scale** and **Move** these pieces to suit your joints.



New surfaces scaled and positioned

These surfaces should be positioned as shown below.



New surfaces

- **Select** the surfaces and press the **3** key to increase their surface smoothness.
- Select **Edit → Delete by Type → History**.

3 Place a lattice on the cube

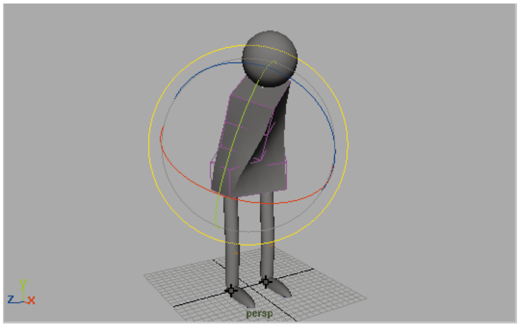
- **Select** the cube. In object mode, you will pick one of the cube's faces.
- Press the **up arrow** to go to the root *nurbsCube* node.
- Select **Deform → Create Lattice**.

4 Bind the lattice and the sphere

- With the lattice already selected, press the **Shift** key and **select** the head sphere.
- Press the **Shift** key and select the root node of the spine skeleton.
- Select **Skin** → **Bind Skin** → **Rigid Bind**.

5 Test the binding

- Select one of the middle nodes of the spline skeleton.
- **Rotate** to test the quality of the bound surfaces.



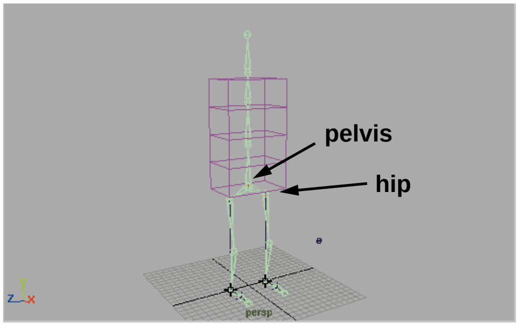
Rotated spine joints

- When you are finished, rotate the joint back to 0 in all three directions.

6 Connecting the legs to the body

- Select **Show** → **Surfaces**.
- Select the right leg's *hip* joint.
- Press the **Shift** key and select the bottom *pelvis* joint of the spine skeleton.
- Select **Skeletons** → **Connect Joint** - ❐ and set **Mode** to **Parent Joint**.
- Click **Connect**.
- Repeat for the other leg, then **Close**.

Now the legs are connected to the spine chain.



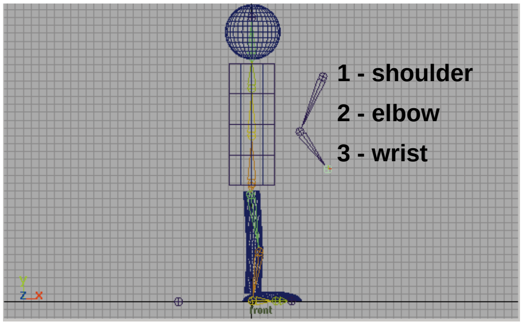
Connected joints

Building arms

You will now build arms for your character. The arms will be built out of cylinders that will again require a joint flexor to handle the bending in the elbow. You will also add a Sculpt object flexor to the upper arm to put some bulging into the arm.

1 Create skeleton joints

- Select **Skeletons** → **Joint Tool**.
- In the Front view, place 3 joints off to the side as shown below. When you finish, rename the joints as labelled:

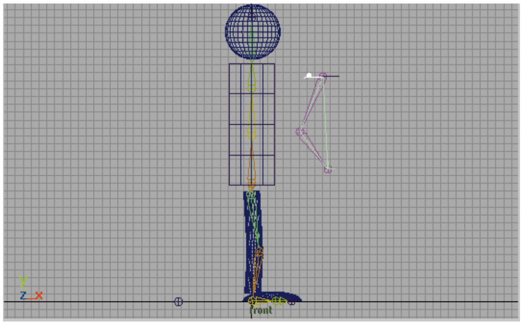


Arm joints

- Press **Enter** to finish.

2 Add an IK chain

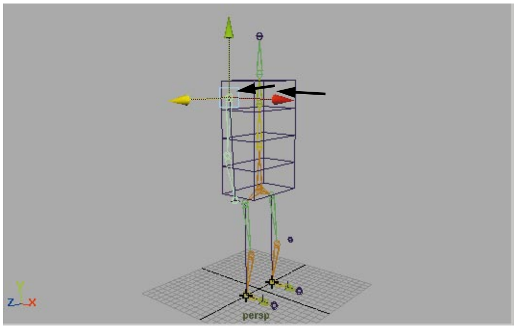
- Select **Skeletons → IK Handle Tool**.
- Click on the *shoulder* joint.
- Click on the *wrist* joint to create an IK chain.



Arm IK chain

3 Move the arm into place

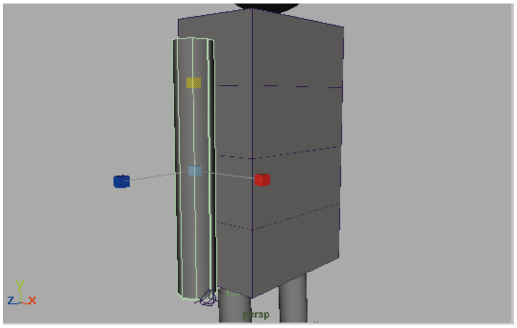
- **Select** the *shoulder* joint.
- In the Front view, **Move** the arm back to the body.
- In the Side view, **Move** the arm to the side of the body.



Positioned arm

4 Create a cylindrical surface

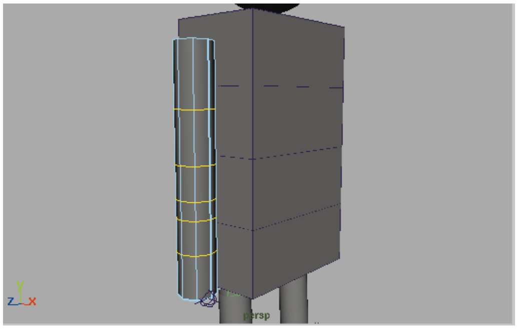
- **Select** the arm's IK handle.
- **Move** it down to straighten the arm.
- Select **Create → NURBS Primitives → Cylinder**.
- In the Channel box, rename this surface to *rightArm*.
- **Scale** and **Move** the sphere to until it covers the arm joints of the skeleton.



Arm cylinder

5 Insert isoparms

- Use the same method you learned with the leg to **Insert** isoparms onto the arm to divide the surface.



Inserting isoparms

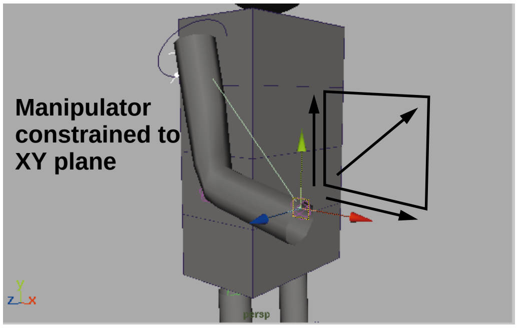
6 Bind the surface

- With the cylinder already selected, press the **Shift** key and **select** the *shoulder* joint.
- Select **Skin** → **Bind Skin** → **Rigid Bind.**

7 Test the motion

- Select the **Move** tool.
- In the Perspective view, press the **Ctrl** key on the Z-axis handle.

  Now the middle manipulator is constrained to move only along the X- and Y-axis plane.
- **Click-drag** on the center manipulator to test the arm.



Testing the motion

- When you are finished, return the arm to the character's side.

Tuck and bulge

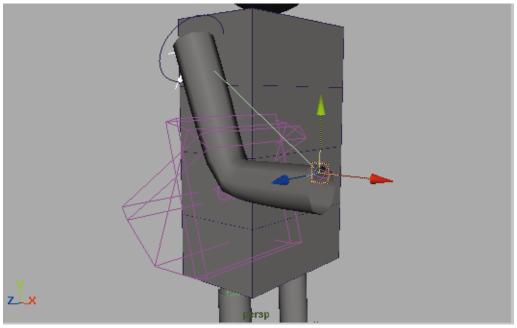
You will now use flexors to set the tuck of the elbow using a lattice and the bulge of the upper arm using a sculpt object deformer. You will then use Set Driven Key to link the rotation of the arm to the scale of the sculpt object.

1 Create a joint flexor

- **Select** the *elbow* joint.
- Select **Skin** → **Edit Rigid Skin** → **Create Flexor**...
- In the Create Flexor window, set the following:

  Flexor Type to **lattice**;

  Joints to **At Selected Joints**.
- Click **Create** then **Close**.
- **Select** and **Move** the IK handle to test the deformation of the elbow area.



Joint flexor

- To create a sharper elbow, select the flexor and, in the flexor input node in the Channel box, set:

  Rounding to **10**.

2 Create a muscle flexor

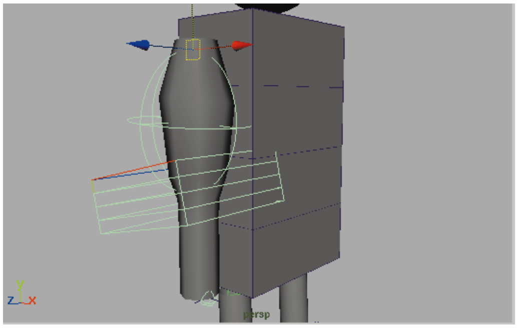
- Move the arm to an extended position beside the body.
- Select the *shoulder* joint.
- Select **Skinning → Create Flexor...**
- In the Create Flexor window, set the following:

  Flexor Type to **sculpt**;

  Bones: **At Selected Bone** to **On**;

  Joints: **At Selected Joint** to **Off**
- Click **Create** and **Close**.

  A sculpt object is placed on the cylinder and is deforming the surface to create a bulge in all directions.

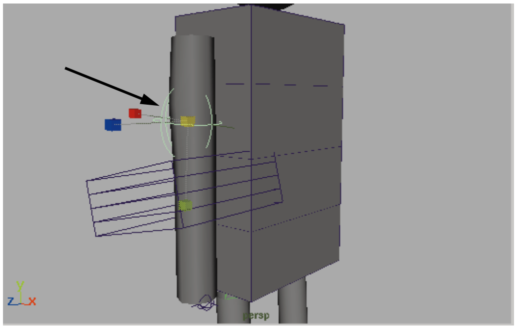


Bone flexor

3 Scale the sculpt object

When the arm is straight, you don't want it to be bulging. You will therefore scale it down to minimize the deformation.

- **Select** the *sculpt1Sphere* node.
- **Scale** it down along the X and Z axes.



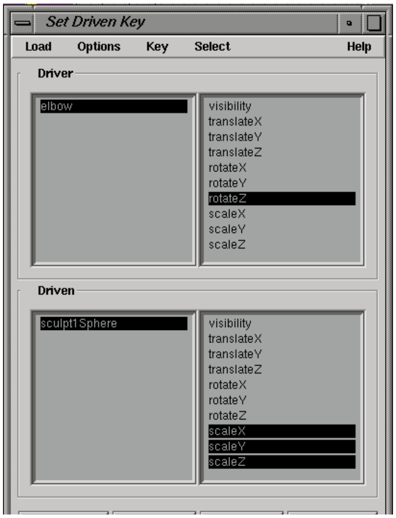
Scaled sculpt object

4 Set Driven Key

You will now drive the shape of the sculpt object with the rotation of the *elbow* joint.

- Select **Animate → Set Driven Key → Set - ❐**.
- Click on **Load Driven** to load the sculpt object.
- Click on the **scale X**, **Y** and **Z** attributes from the right column.

  These are the attributes that you want to drive.
- Select the *elbow* joint.
- In the Set Driven Key window, click on **Load Driver**.
- Click on **rotate Z** in the right column, and click the **Key** button.



Driver

elbow

visibility
translateX
translateY
translateZ
rotateX
rotateY
rotateZ
scaleX
scaleY
scaleZ

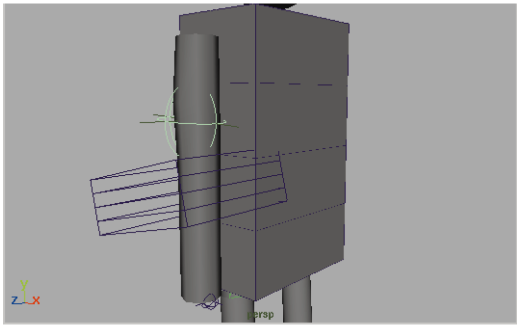
Driven

sculpt1Sphere

visibility
translateX
translateY
translateZ
rotateX
rotateY
rotateZ
scaleX
scaleY
scaleZ

Set Driven Key options

Now the current rotation setting for the elbow is linked to the current scaling of the sculpt object.



Set Driven Key start position

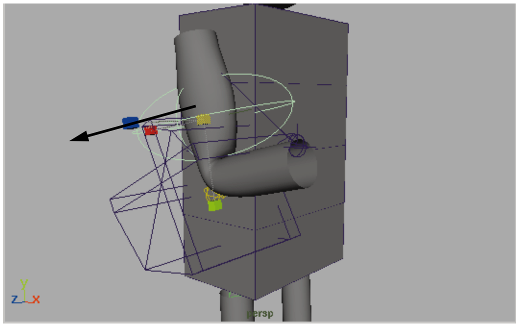
5 Set a second key

- **Select** the hand's IK handle.
- **Move** the handle to raise the arm.

This automatically rotates the *elbow* joint.

- Select the *scultp1Sphere* node.
- **Scale** it out along the X-axis.
- Click on **Key** button.

This sets the finish position for the Set Driven Key.

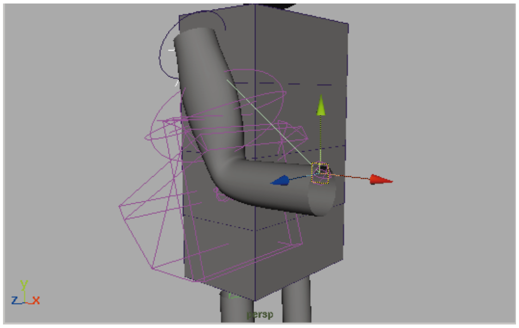


Set Driven Key finish position

- Close the Set Driven Key window.

6 Test the results.

- **Select** the hand's IK handle.
- **Move** the handle to watch the upper arm bulge.



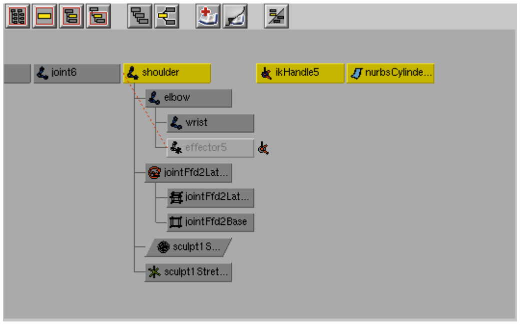
Arm motion

Duplicate the arm

Now just like the leg, you will need to duplicate the arm, the IK handle, the surface, and any connected nodes.

1 Select the parts

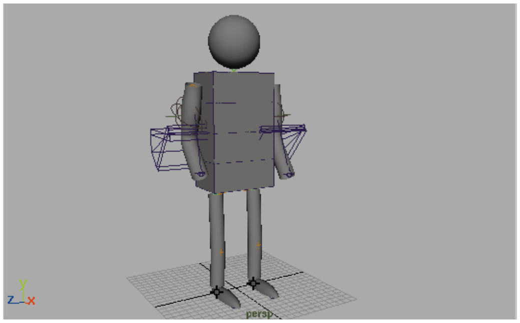
- In the Hypergraph, **Select** only the skeleton and IK handles.



Selected nodes

2 Duplicate

- Select **Edit** → **Duplicate** - ❐ and make sure that **Duplicate Upstream Graph** is set to **On**.
- Click **Duplicate** then **Close**.
- **Move** the new group to the character's left side.

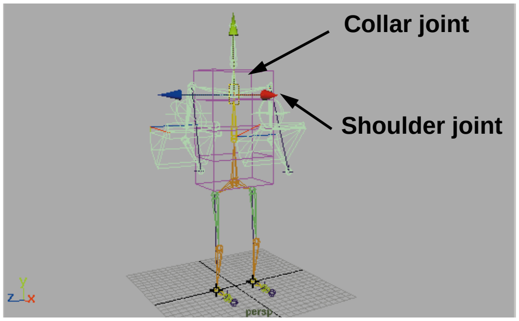


Duplicated arm

3 Connect the joints

- **Select** the right arm's *shoulder* joint.
- Press the **Shift** key and select the bottom *collar* joint of the spine skeleton.
- Select **Skeletons** → **Connect Joint** - ❐ and set **Mode** to **Parent Joint**.
- Click **Connect** then **Close**.
- Repeat for the other arm.

Now the arms are connected to the spine chain.



Connected joints

4 Save your work

Conclusion

You now have a biped character all hooked up for a stroll. You can roll the feet and deform the surfaces using the skeleton to control the motion.

While this character doesn't seem very sophisticated, the binding and deformation techniques would be very similar in a more complex model. Your cylindrical arms could easily be sculpted surfaces while the binding and flexors would be similar.

In the next lesson, your Primitive Man will evolve into a walking figure. This will involve setting keys on the IK handle groups and some of the joints.

17 Animating a Walk Cycle

You have now built a character who can be animated walking and moving his arms. In this lesson, you will animate a full walk cycle complete with body movements and hand gestures.

Primitive Man on the move

To animate the walk cycle, you will build up the motion one piece at a time. When you start, the motion will seem mechanical. However, once you layer in the roll of the feet, the twist of the pelvis, and the rotation of the spine, Primitive Man will be walking with more character.

In this lesson you will learn the following:

- How to animate the forward movement of the character
- How to animate the feet going up and down
- How to animate the twist of the character's pelvis
- How to animate the roll of the foot
- How to animate the character's torso and arms
- How to animate a two-node camera

Initial set-up

For this lesson, you will animate Primitive Man walking. You should start with the file you saved in the last lesson.

1 Display selection handles

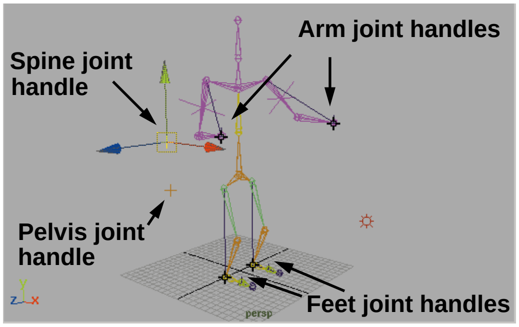
- In Perspective view, select **Show** → **Surfaces** and **Show** → **Deformers** to turn these off.
- **Select** the *pelvis* joint, the *spine* joint, the left arm IK handle, and the right arm IK handle.
- Select **Display** → **Object Components** → **Selection Handles**.

2 Lower the pelvis

- **Select** the *pelvis* joint and lower it until Primitive Man's knees are slightly bent.

3 Edit the handle positions

- Press **F8** to go into component mode.
- **Select** the selection handle pick mask.
- **Move** the pelvis and the spine handles so they are outside and behind the character.



Selection handles labelled

- Press **F8** to go back to object selection mode.

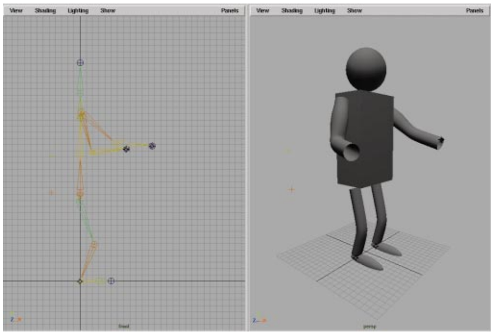
4 Set up your view panels

- Set up a Perspective view panel and a Front view panel.
- In the Front view, select **Show** → **None** then **Show** → **Handles** and **Show** → **Joints**.

This panel will be used to watch the movement of the joints.

- In the Perspective view, select **Show** → **None,** then **Show** → **Surfaces,** and then **Show** → **Handles.**

This panel will be used to watch the movement of the surfaces.



View panel layout

Animate the pelvis

To create a walk cycle, the beginning and end point of the forward movement of the entire skeleton should be established. This is done by keying the positions of the skeleton's pelvis joint.

1 Set a key on the pelvis

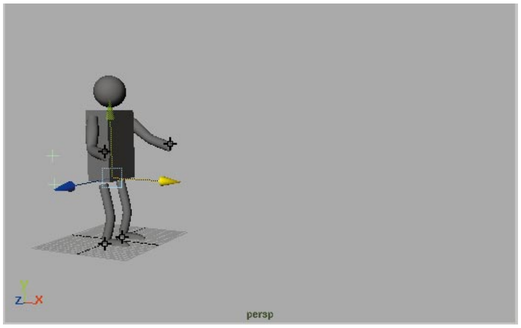
- Open the Animation Preferences window, and set the following in the Keys section:

  Default In Tangent to **Linear**;

  Default In Tangent to **Linear.**

  This will set your tangents in the animation curve to react in a more rigid fashion. You will refine these later in the lesson.
- Go to the **Animation** menu set.
- **Select** the skeleton's *pelvis* joint.
- Go to frame **1**.
- In the Channel box, click on the **Translate X** channel to select it.
- Click with your right mouse button and select **Key Selected** from the menu.

  For now, you are only going to key the forward motion of the character.



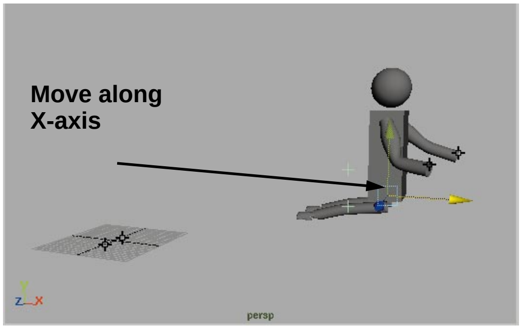
First pelvis position

2 Move pelvis and set a second key

You will now key the end location of the pelvis to create the linear motion of the skeleton body.

- Go to frame **90**.
- In the Front view, **Move** the *pelvis* joint about **60** units along the X-axis.

  The pelvis moves while the character's feet try to stay in their initial position. You will correct the position of the feet later.
- Set another key on the **Translate X** channels.



Second pelvis position

Animate the feet sliding

You will now key the horizontal positions of the feet to establish their forward movement, in relation to the forward movement of the pelvis. This will result in a sliding motion on the feet.

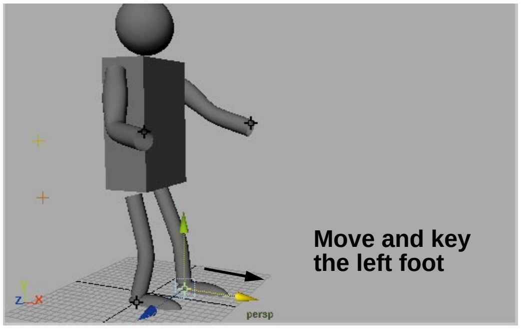
1 Position feet and set a key

You will key the starting position of the two feet in the position of a full stride.

- Go to frame **1**.
- In the Perspective window, select the foot selection handle for the left leg.
- **Move** the handle forward along the X-axis.

The joint at the knee should not flatten and the ankle joint should remain with the pivot point.

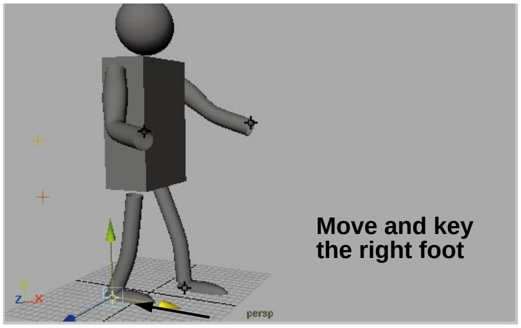
- Press **Shift w** to set a key on all the *Translate* channels of the group.



Left leg position

- Select the right IK handle foot group.
- **Move** the right group back along the X-axis.

  The skeleton should be in a walking position as shown below.
- Press **Shift w** to set a key on translate channels.



Right leg position

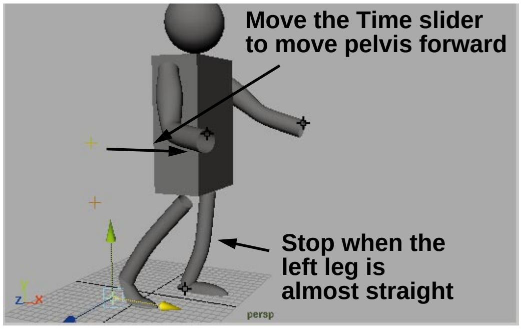
2 Move Time slider

You will now use the forward movement of the pelvis to help you place your joints.

- Slowly move the current frame indicator in the **Time slider** forward.

The left leg straightens as the pelvis moves forward and the right leg separates from the IK handles.

- Continue to move the **Time slider** until the left leg has almost straightened.

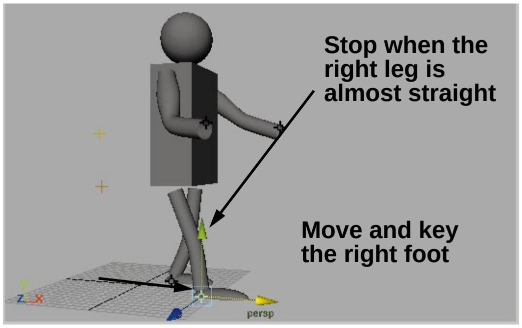

Move the Time slider to move pelvis forward
Stop when the left leg is almost straight
persp

Time slider moved to frame of first step

3 Move right foot and set keyframe

You will now move the right foot so that the right leg is in front of the body.

- **Move** the handle to the right.
- Position the foot at the end of the first step. The leg should be almost straight.
- Press **Shift w** to set translation keys.


Stop when the right leg is almost straight
Move and key the right foot
persp

Right foot moved to first step

4 Pick left foot and set a key

A key *must* now be set on the left foot so that it remains planted while the right foot is moving.

- **Select** the left foot using its selection handle.
- Press **Shift w** to set a key.

Note: This key is *very* important to ensure that the one foot stays on the ground while the other foot moves.

5 Move Time slider

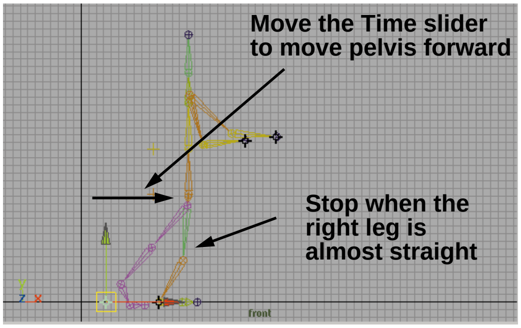
You will now move the Time slider to the frame that corresponds to the end of the second step.

- Slowly scrub forward in the **Time slider**.

  Watch the movement of the skeleton.

- Stop at the frame when the right leg has almost straightened.

  This should be the position of the back right leg at the end of the second step.


Move the Time slider to move pelvis forward
Stop when the right leg is almost straight
front

Time slider moved to timing of second step

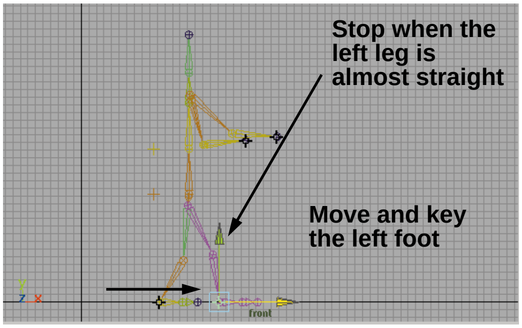
6 Move left foot and set a key

You will position the left foot at the end of the second step.

- **Move** the left foot along the X-axis to position the handle at the end of the step forward.

  Again, the left right leg should be almost straight, with the heel touching the axis.

- Press **Shift w** to set a key.



Stop when the left leg is almost straight

Move and key the left foot

Left foot moved to position of second step

7 Save your work

- Select **File → Save Scene As...** and save your work as *walk01*.

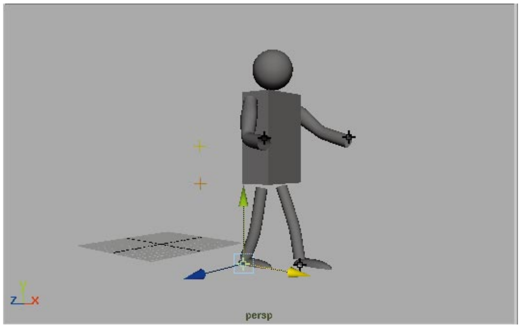
8 Set keys for the remaining steps

Use the same workflow to continue sliding the feet forward while planting the other foot. You will need to set the finished step positions so that the feet keep up with the animated movement of the pelvis. The repeated workflow is as follows:

- **Select** the other foot.
- Press **Shift w** to set a key.

  As mentioned, this set key is *very important* because it plants the foot properly.

- Scrub in the **Time slider** to the end of the character's next step.
- **Move** the foot along the X-axis until it is almost straight.
- Press **Shift w** to set a key.
- Continue these steps to frame **90**.



Completed stepping cycle

Note: Make sure to set keys at the beginning and end of each step for each foot.

Edit the animation curves

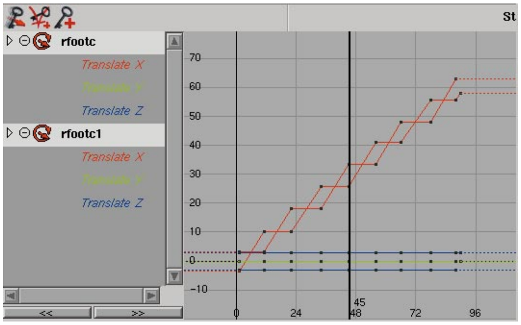
To refine the in-between motion of the feet, you can use the animation curves to view and change the tangent options for the feet.

1 View the curves in the Graph Editor

You will edit the animation curves produced by the keys in the Graph Editor.

- **Select** both of the feet.
- Open a Graph Editor panel.
- In the Graph Editor, select **View → Frame All**.

  The step pattern of the keys should look as follows.



Forward step curves in Graph Editor

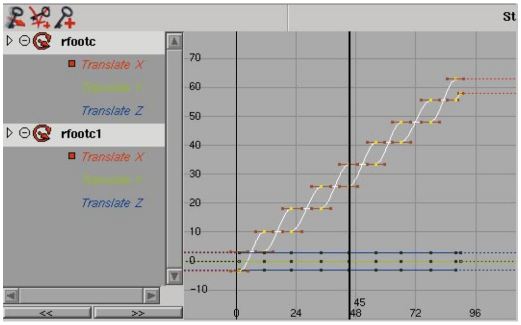
- Play back the animation to see the motion.

  With linear tangents, each foot remains planted while the other moves, but the resulting motion is a little "robotic."

2 Change the curve tangents

The tangent curves can be changed to soften the motion of the walk.

- In the Graph Editor, select the two **Translate X** curves for the feet.
- Select **Tangents** → **Flat**.



Forward step curves using flat tangent

- Play back the animation to see change in the motion.

Animate the feet (up and down)

You will now key the vertical raising and lowering of the feet to establish the stepping action.

1 Turn on Auto Key

You will now use Auto Key to help you with the raising of the feet.

- Click on the **Auto Key** button in the right side of the Time slider to turn it on.
- Open the Animation Preferences window, and set the following in the Keys section:

  Default In Tangent to **Spline**;

  Default In Tangent to **Spline.**

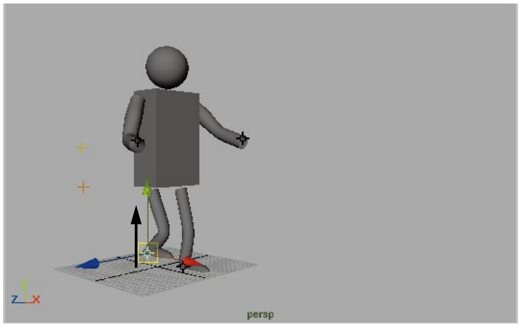
  You now want your curves set more smoothly.

2 Raise the right foot at midstep

Key the high point of the raised foot at the appropriate frame.

- Go to frame **1**.
- **Pick** the IK handle for the right foot.
- Move the **Time slider** to the frame at the midpoint of the first step.
- Click-drag the foot along the Y-axis using the manipulator handle.

  This sets a new key for the Y-axis channel of the foot using Auto Key.



Raised step for right foot

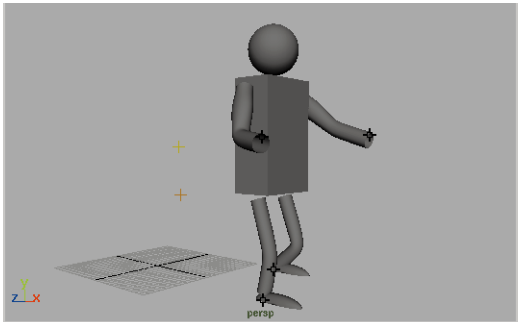
3 Set keys for left and right foot

You will now set keys for the vertical movement of the feet for each step.

- Move the **Time slider** to the midpoint of the next step for the right foot.

  Pass over the full step of the right foot. You will key these steps later.
- **Move** the foot up along the Y-axis.

  Again a key is automatically set.
- Use the same process to keyframe the raised steps of the left foot.
- Play back the results.



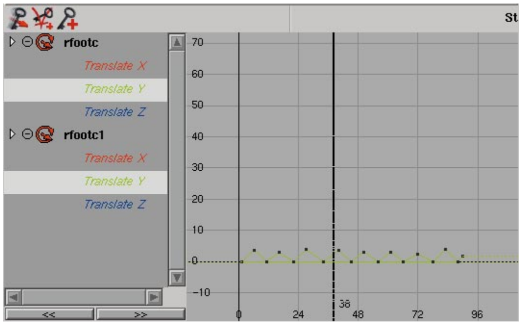
Playback of raised feet

4 View the animation curves

The horizontal and vertical animation of the feet can be viewed in the Graph Editor.

- **Select** both IK handles.
- Press the **Ctrl** key and click on the **Translate Y** channels for both feet.

  This selects only the Translate Y curves.



Step curves in Graph Editor

- Play back the animation in the Time slider.

5 Save your work

You are now going to animate the character's pelvic motion. Afterwards, you will return to the foot to add the heel rotation.

Animate the pelvic rotations

To create a more realistic action, the pelvis' position and rotation will be set to work with each step. You can again set keys for the translation and rotation of the pelvis using Auto Key.

1 Key the Y translation of the pelvis

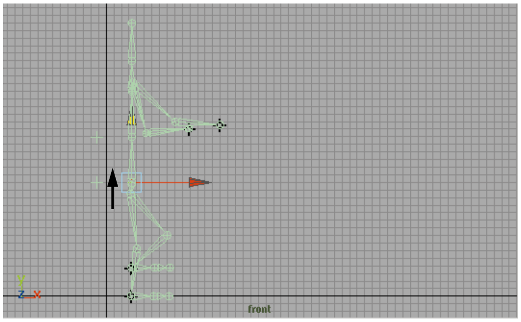
When the character walks, the pelvis will raise as each foot is raised then lower as the both feet touch the ground.

- **Select** the *pelvis* joint using its selection handle.
- Go to frame **1**.
- In the Channel box, click on the **Translate Y** channel name.
- Click with the right mouse button and select **Key Selected** from the menu.

  Now keys can be set on this channel using Auto Key. You must always have at least one key set for Auto Key to recognize a channel.

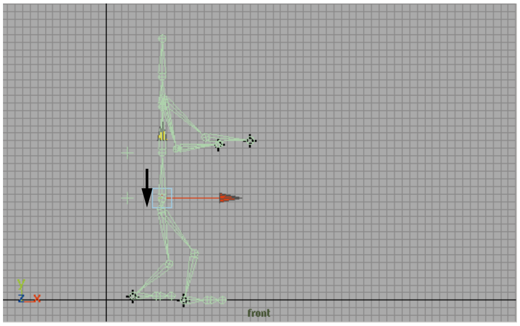
2 Auto Key Y translation of the pelvis

- Move the **Time slider** to the point where the first leg is fully raised.
- Select the **Move** tool.
- **Click-drag** on the Y translate manipulator handle to raise the pelvis slightly. You won't need dramatic movement in the pelvis for the walk to look good.



Raised pelvis

- Move the **Time slider** to the point where both feet are on the ground.
- Click-drag on the Y translate manipulator handle to lower the pelvis slightly.



Lowered pelvis

- Continue raising and lowering the pelvis in a similar manner. Raise it every time either foot is raised and lower it when both feet are on the ground.
- Play back the results.

Note: It would be very easy to write an expression that links the Y translation of the hip to the average of the Y translation of the two feet. You haven't done this since an expression might give a robotic motion. By using interactive transformations – then setting your own keys – you will introduce a randomness that will seem more natural.

3 Key the rotation of the pelvis

You will now animate the pelvis rotation to give the walk a little more motion.

- Go to frame **1**.
- In the Channel box, click on the **Rotate X, Y** and **Z** channel names.

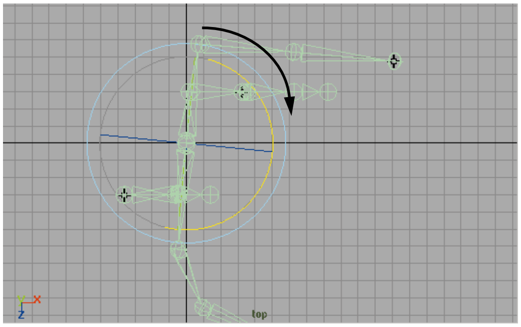
- Click with the right mouse button and select **Key Selected** from the menu.
- Select the **Rotate** tool.

4 Auto Key rotation in Top view

You will now key the rotation in the Top view using Auto Key, which is still on.

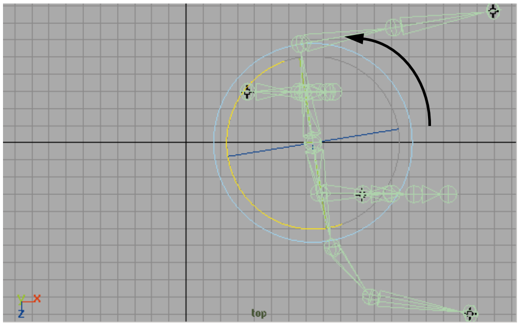
- In the Top view, **Rotate** the pelvis, using the rotation handle, towards the right foot which is currently behind the body.

  Another Auto Key is set at this pose.



Rotate pelvis toward right foot

- Move the **Time slider** forward until the left foot is behind the body.
- **Rotate** the pelvis in the opposite direction to rotate the body towards the left foot.



Rotate pelvis toward left foot

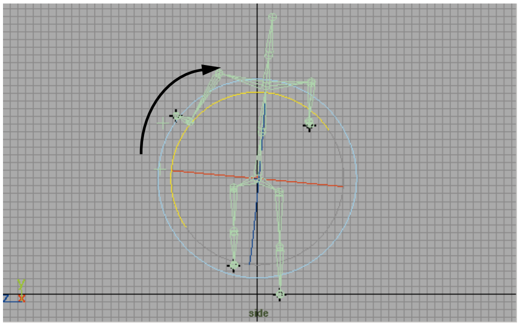
5 Set more Auto Keys

- Repeat these rotations keeping in mind that they should always orient the pelvis towards the lagging foot.
- Play back the motion.

6 Auto Key rotation in Side view

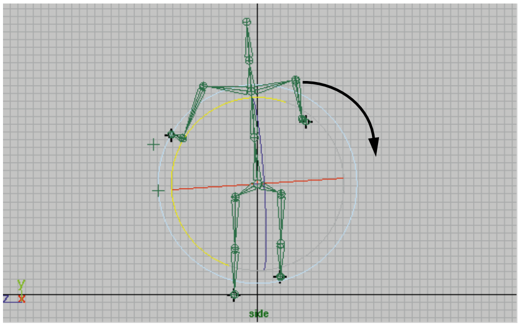
- Go to frame **1**.
- Move the **Time slider** to the point where the first leg is fully raised.
- In the Side view, **Rotate** the pelvis using the rotation handle to rotate the hips so that the right hip is raising with the right leg.

Another key is set at this pose.



Rotated pelvis with right foot raised

- Move the **Time slider** forward until the left foot is in front of the body.
- **Rotate** the pelvis in the opposite direction to rotate the hips up as the left foot raises.



Rotated pelvis with left foot raised

7 Set more keys

- Repeat these rotations to always orient the pelvis with the raised foot.
- Play back the motion.

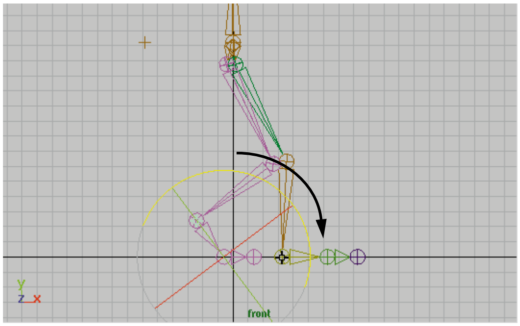
8 Save your work

Animate the heel rotation

In the last lesson, you spent a great deal of time preparing the foot for the *heel to toe* motion that a foot goes through when walking. You are now going to use Auto Key to keyframe the foot rotations to take advantage of your work.

1 Set a key on the right foot's roll

- Go to frame **1**.
- **Select** the right foot using its selection handle.
- **Rotate** the foot forward until it is pivoting around the ball of the foot.
- In the Channel box, click on the **Rotate Z** channel name.
- Click with the right mouse button and select **Key Selected** from the menu.

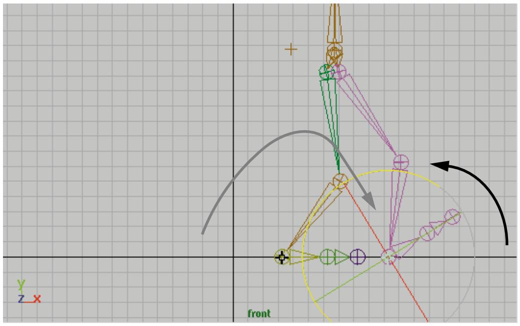


Foot rotated forward

2 Auto Key the rotation

- Move the **Time slider** forward until the right foot has raised and has just lowered back to the ground.
- **Rotate** the foot back until it is pivoting around the heel of the foot.

Auto Key will set a second key for you.



Foot rotated back

- Now move the **Time slider** to the point where the right foot is about to raise from the ground.
- **Rotate** the foot forward until it is again pivoting around the ball of the foot.
- Repeat these steps until the rotation of the right foot is complete.

3 Set a key on the left foot's roll

- Go to frame **1**.
- **Select** the left foot using its selection handle.
- **Rotate** the foot back until it is pivoting around the heel of the foot.
- In the Channel box, click on the **Rotate Z** channel name.
- Click with the right mouse button and select **Key Selected** from the menu.

4 Auto Key the rotation

- Move the **Time slider** to the point where the left foot is about to raise from the ground.
- **Rotate** the foot forward until it is pivoting around the ball of the foot.
- Move the **Time slider** forward until the left foot has raised and has just lowered back to the ground.
- **Rotate** the foot back until it is again pivoting around the heel of the foot.
- Repeat these steps until the rotation of the right foot is complete.

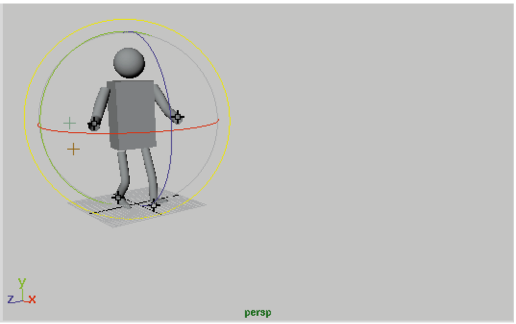
5 Save your work

Setting keys for the spine

As Primitive Man walks along, you want his body to look up and down as if he is a tourist exploring an unfamiliar place. You will keyframe this motion using the rotation of the spine joint that you set up earlier with a selection handle.

1 Set keys on the joint rotation

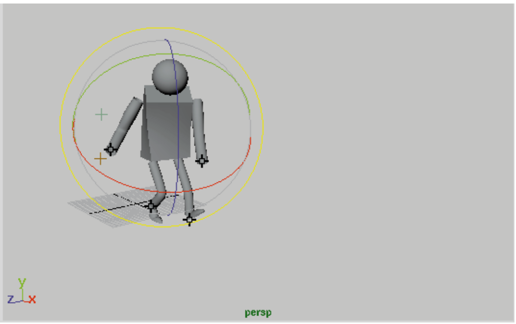
- Go to frame **1**.
- **Select** the *spine* joint using its selection handle.
- Press **Shift e** to set keys on the three rotation channels of this node.



Selected spine joint

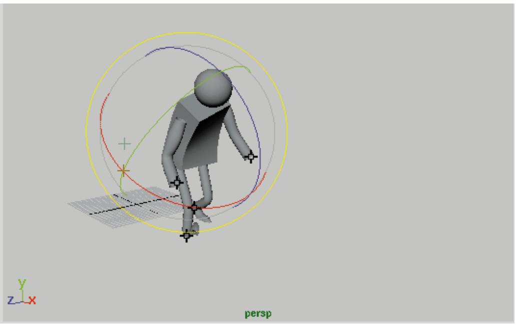
2 Auto Key the rotation of the joint

- Go to frame **22**.
- **Rotate** the *spine* joint so that the character appears to be looking down to his right.



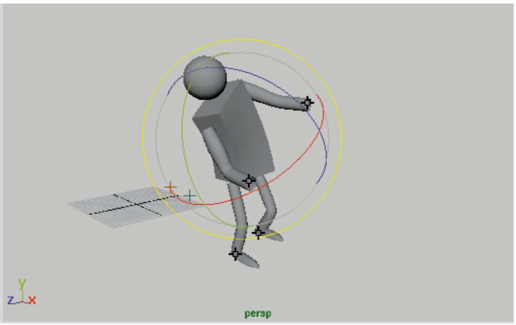
Pose at frame 22

- Go to frame **36**.
- **Rotate** the *spine* joint so that the character appears to be looking down and to his left.



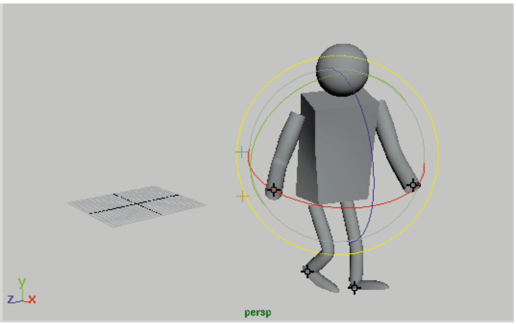
Pose at frame 36

- Go to frame **61**.
- **Rotate** the *spine* joint so that the character appears to be looking up and to his left.



Pose at frame 61

- Go to frame **90**.
- **Rotate** the *spine* joint so that the character appears to be looking ahead in a normal position.



Pose at frame 90

3 Save your work

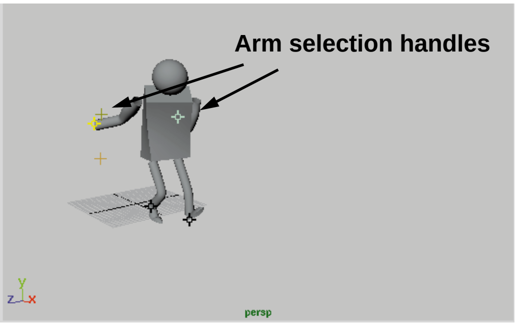
Keying the arm motion

As Primitive Man walks, he will likely be moving his arms to make gestures. You will now set keys on his arms to add these to the animation.

1 Key the arm IK handles

- **Select** the two arm IK handles using their selection handles.

- Press **Shift w** to set keys on their translation channels.
- Move the **Time slider** forward.
- Oops! The arms seem trapped. This is because you locked their position in world space. You need to find a way to make the arms work with the body.



Arms locked to keyed position

2 Group the arm IK handles

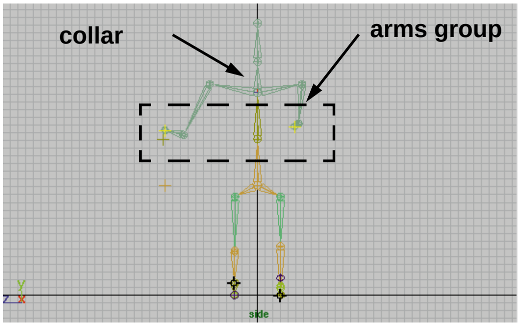
You will now group the handles so that they can be moved together.

- Go to frame **1**.
- With the two handles selected, select **Edit → Group**.
- Rename the group as *arms*.

3 Parent the group to the shoulder joint

To make the arms move with the body, you will parent the new group to the skeleton's shoulder joint.

- Press the **Shift** key and select the *collar* joint of the skeleton.



Parenting arm group to shoulder joint

- Select **Edit → Parent** - ❐ and in the option window, make sure that **Preserve Position** is set to **On**.
- Click **Parent** and **Close**.
- Move forward in the **Time slider**.

Now the arms move with the body even though there are keys set on the handles. These keys will always remain under the *arms* node.

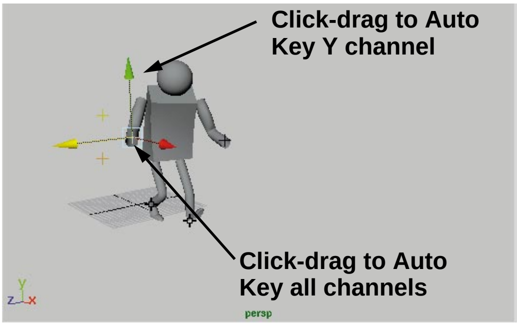
4 Auto Key the hand positions

You can now set keys on the arm IK handles and the resulting positions will remain with the body.

- Move the two handles to set up positions like those shown below.

Auto Key will set keys for the positions as you move the handles. Remember that Auto Key only sets keys for manipulators that you use.

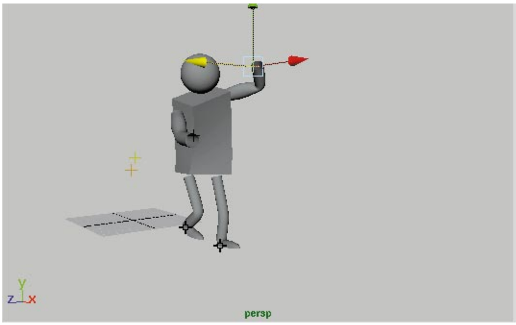
Therefore, if you drag on the Y-axis handle, that is the only key set. If you want keys set for all three translation channels, then click-drag on the middle box manipulator which controls all the axis.


Click-drag to Auto Key Y channel
Click-drag to Auto Key all channels
Y
Z
X
persp

Possible arm position

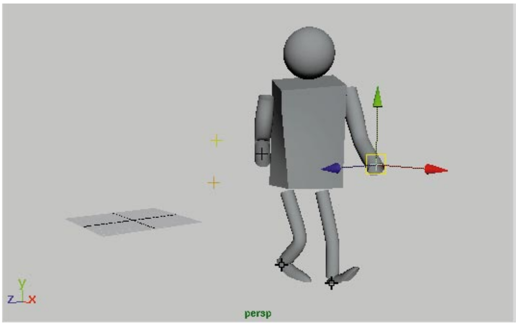
Note: If you are having trouble getting the arm to raise above Primitive Man's head, or if you experience flipping, read the Rotate Plane IK Solver section on the next page.

As you move the arms, watch to make sure that you are not intersecting the torso cube.



Possible arm position

Use all three axis handles to help with the positioning.



Possible arm position

5 Turn off Auto Key

- When you are finished, turn off **Auto Key**.

The rotate plane IK solver

When moving the arm IK handles, you will notice that you seem to have limited control over how the elbow moves. Also, at some point in your animation, the whole arm may flip to a different orientation. Since you created this IK handle using the *rotate plane solver*, you can use more manipulator controls to define how the whole arm works.

When you set up a *rotate plane IK handle*, the whole length of the chain is controlled by a plane that is defined by the *handle vector* which runs between the *root joint* and the *end effector*, and a secondary vector that is called the *pole vector*. By default, the IK handle will manipulate the chain so that it works within this plane.

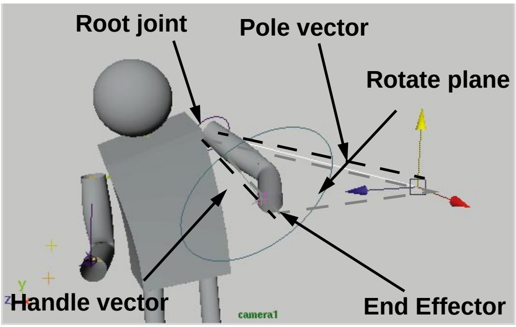


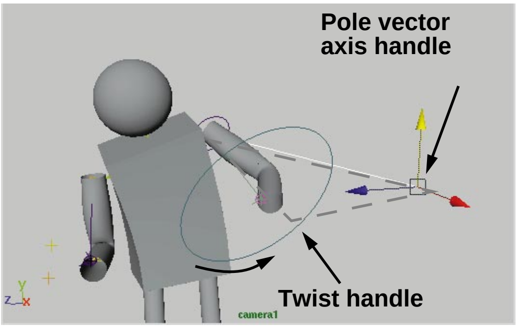
Diagram of rotate plane

To get more freedom in the motion of the IK handle, you can redefine the plane's orientation using special manipulator handles.

How to work with rotate plane manipulators

The rotate plane attributes can be set in the Channel box or you can use the rotate plane manipulators.

- **Select** the IK handle for the arm.
- Select the **Show Manipulator** tool.
- You now get a few manipulator handles to help you control the arm.



Rotate plane manipulator handles

The **pole vector axis** handle lets you manipulate the whole rotate plane. By positioning the handle, you change the orientation of the plane.

The **twist** handle lets you orient the arm off of the plane. This lets you move the elbows out away from the plane.

How to avoid flipping in the arm

Flipping occurs when the end effector is moved through the plane. If you experience flipping, you can use the **pole vector axis** handle to move the plane out of the way. You may need to set keys on this handle to control flipping in a more complex arm motion.

Try these handles to see how they help you position the arm. You can then **Key Selected** in the Channel box on the associated attributes.

Animating a two-node camera

You will now add a new camera to the scene and animate it so that you can follow Primitive Man as he walks.

Cameras can be created with one, two or three nodes. Most cameras only need one node which lets you key the camera's position and rotation. A two-node camera gives you nodes for both the *camera point* and the *view point*. Both these nodes can be keyed individually.

1 Set up your panel display

- In the Perspective view, select **Show** → **Cameras** and **Show** → **Pivots** to add these objects to the display.

You will need to see these in order to work with the camera.

2 Create a two-node camera

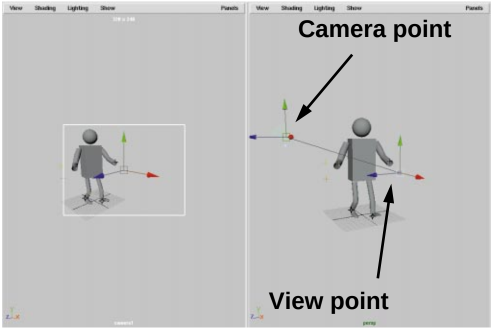
- Select **Create** → **Create Camera** - ❐.
- Scroll down to the **Animation Options** section and set **Nodes** to **Two**.
- Click **Create** then **Close**.

3 Position the camera

- Select the **Show Manipulator** tool
- In the Perspective view, position the *camera* and *view point* handles to set up the camera.
- In the other view panel, select **Panels** → **Perspective** → **camera1**.
- From the *camera1* view, select **View** → **Camera Settings** → **Resolution Gate**.

You can now position the camera using both of these panels. Put the *view point* of the camera in front of the character.

- In the Perspective view, select **Show** → **Surfaces** then shade the view.

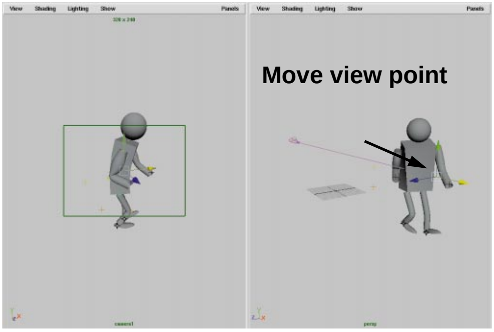


Camera manipulator handles

4 Set keys on the view point

You will now set keys on the view point to follow the character from frame 1 to 90.

- Go to frame **1**.
- Select the **Select** tool.
- **Select** the view point on its own.
- Press **Shift w** to keyframe the position of the view point.
- Go to frame **90**.
- **Move** the view point so that it is again in front of the character.
- Press **Shift w** to keyframe the position of the view point.



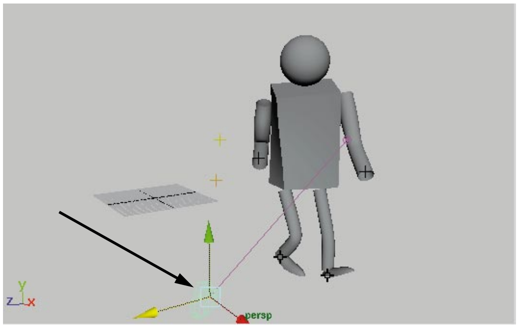
View at frame 90

5 Set keys on the camera node

The camera animation is now pointing at Primitive Man, but he is not always framed in the view as well as you'd like. You can set keys on the eye point node to fix this.

- Select the *camera* node by clicking on the camera icon.
- Go to frame **1**.
- Press **Shift w** to set keys on the translation channels of the *camera1* node.
- Go to frame **90**.

Move the *camera* node in the Perspective view to the front of the character to frame the view.



Camera point position at frame 90

- Press **Shift w** to set keys on the translation channels of the *camera1* node.
- Play back the results.

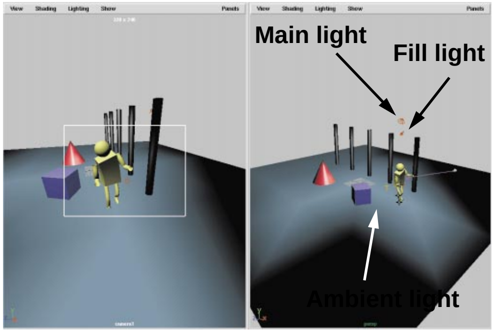
If you don't like the framing in the in-between frames, then you can reposition the camera at either frame 1 or 90 and set new keys. Repeat this until you get the camera movement you want.

Props, color and lighting

On your own, build some props, add color materials to the surfaces, and then light the scene. You will complete these steps on your own to build the set that you want, with the lighting that you feel suits the walk cycle.

In the example shown below, several columns were built to accentuate the length of Primitive Man's walk and to give a frame of reference for his walking. At this point, you may want to edit your camera animation to suit your new scene.

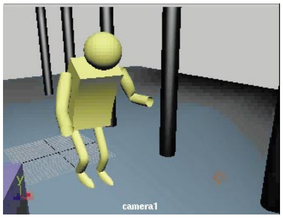
Three lights were used to illuminate the scene. These include one spot light with shadows on and an intensity of 1.0, which is the main light placed above Primitive Man and to his right. A second light with an intensity of about 0.4 is used as a fill light and is placed up and towards Primitive Man's left. An ambient light with an intensity of 0.1 further fills in dark spots and is placed near the ground. You can play with your own three lights to illuminate the scene.



Props displayed with hardware lighting

Testing the motion

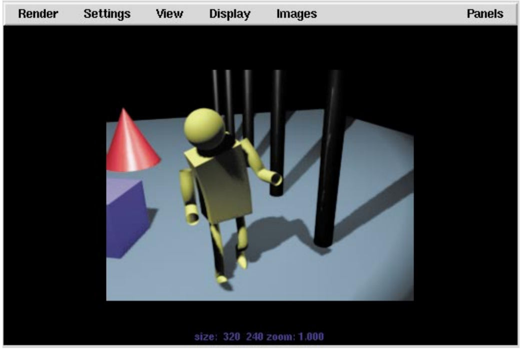
You can now Playblast the scene to test the motion. This will give you the chance to confirm the location of your props and whether or not you like the camera animation.



Playblast of animation

Rendering the animation

You can now render the scene. To render, you should consider the various issues taught throughout this book. You need to set attributes on the surfaces themselves and in the Render Globals. Listed below is a checklist of some of the issues you should consider when rendering. Test them out when you render the Primitive Man scene.



Software render test

Note: You will have encountered most of these issues in earlier **Learning Maya** projects. This checklist offers a compiled list of those issues and some new ones.

OBJECT ISSUES

Some render attributes need to be set for your objects' *shape* nodes. You can set these attributes in the **Render Flag** window, in the *shape* node's **Render** section in the Attribute Editor, or in the **Attribute Spread Sheet** window. Below are some of the attributes you should consider when you render:

✔ Surface Tessellation

Set a NURBS surface tessellation that is appropriate to the scene. Larger and more prominent objects will require a larger tessellation than background elements. It is very important that you don't over-tessellate.

You can also use the secondary tessellation controls to create a more dynamic tessellation suitable for animation. The Use Minimum Screen option let you set a pixel square (default 14 pixels) where your objects tessellation is generated by comparing the screen size of the surface to the pixel square to decide on an appropriate tessellation for the surface.

✔ Reflections and refractions

If you plan to raytrace, remember that you must decide which objects can be reflected by other objects. It is very important not to forget this step and render a scene that doesn't show any reflections.

✔ Motion blur

While you turn on motion blur in the Render Globals, you can also decide which objects will or will not use motion blur. If you have objects that are not moving or are barely moving, turn motion blur off to speed up rendering.

✔ Lights and shadows

Limit the number of lights casting shadows in your scene. If possible, use depth maps shadows which are a little faster. If you want to add a lot of lights to a scene then consider linking some of the lights to only those objects which need the illumination.

RENDER GLOBAL ISSUES

✔ Turn animation on!

If you want to render an animation, you must turn Animation on in the Render Globals. It is very easy to forget this and send off what you think is a long night of rendering frames, only to come in the next day to just a single frame.

✔ Renderable camera

Do you have the right camera set up for rendering? It is very easy to leave the default *persp* camera as *renderable* when in fact you want to render another camera.

It is also easy to forget to turn *renderable* off for the *persp* camera when you choose the second camera. If you forget, then you end up rendering both cameras which means lots of rendered frames that you don't need.

✔ Masks and depth masks

If you plan to composite your renderings later, you may want to use these settings to create a matte layer (mask) to assist in the compositing process.

✔ Render resolution

What is the render size that you want? Be sure that if you change the pixel size, then use the *resolution gate* in your view panel to make sure that the framing of your scene is preserved.

✔ Raytrace

Do you want to raytrace some of your objects? Remember that Maya has a selective raytracer and only objects that require reflections, refractions or shadows will be raytraced.

Therefore, if you limit your reflective and refractive materials to key objects, then you can raytrace them knowing that other objects in the scene are using the A-buffer.

If you are raytracing, try to limit the number of reflections set in the globals. A setting of 1 will look good in most animations unless, for example, you have a chrome character looking into a mirror.

✔ Render quality

You may want to use the *Recommended Defaults* pop-up to suggest render quality options until you are familiar with the settings.

OTHER RENDERING ISSUES

Here are some other key rendering issues to consider.

✔ Test render, test render, test render

Don't start a major rendering unless you have test rendered at various levels. Render the whole animation at a low resolution with low quality settings to test the motion. Render random full-size single frames to confirm that materials, lights and objects are rendering properly.

The more you test render, the less time you spend re-doing renderings that didn't work out the way you wanted.

✔ Command line rendering

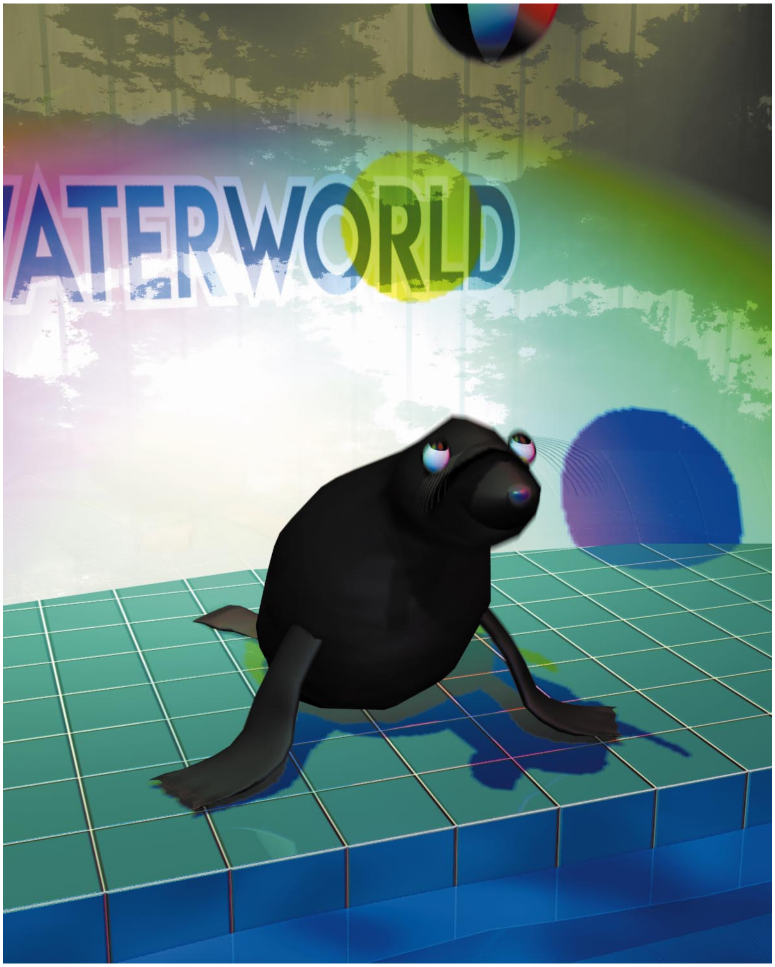
You have learned how to batch render from within Maya. You can also render from the MS-DOS command line. Here is the basic workflow:

- Set up your Render Globals.
- **Save** your scene file.
- Type `Render -help` for a list of all the command line options.
- `chdir` into the directory with your file.
- Enter the `Render` command along with any flags followed by the file name, such as the start and end frames for the rendering as shown in the following:

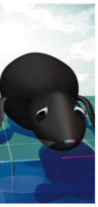
```
Render -s 1 - e 24 renderTest.mb
```

Conclusion

Congratulations! You have now completed your first walk cycle. Now you are going to animate Salty the seal. This final project takes you through a more complex character setup that includes the creation of special controls for Salty's eyes.

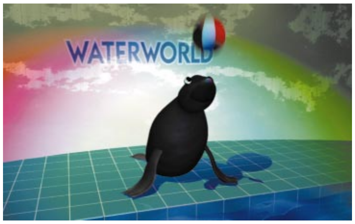


Project Five

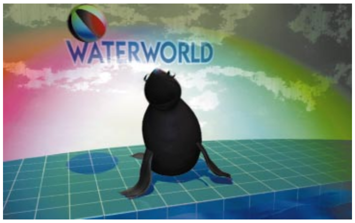


Project Five

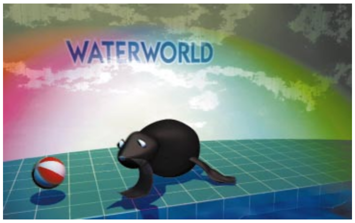
In this project, you are going to model, texture and animate a complete character – Salty the seal. Salty will be built out of primitive shapes that will then be bound to skeleton joints and animated. This lesson integrates modeling, rendering and animating so that the lines between them begin to blur.



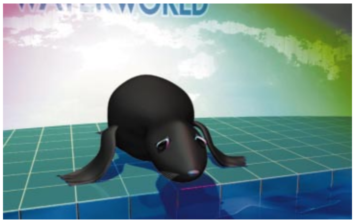
1. Salty begins by bouncing a ball on her nose, two times with ease.



2. Suddenly, the ball is blown to the side of the stage and Salty loses control.



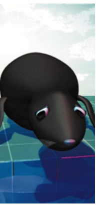
3. Concerned, Salty follows the ball as it bounces out of her reach.



4. Dejected, Salty looks back at the audience unhappy that the ball got away.

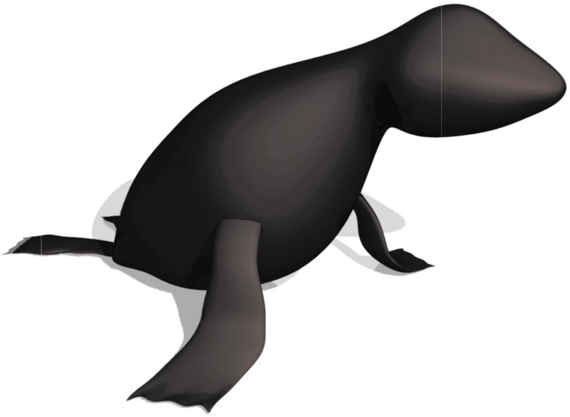
STORYBOARD

PROJECT FIVE



18 Building Salty

This lesson shows you how to build your first Maya character – Salty the seal. Starting with a primitive sphere, you will edit the surface's control vertices to give the sphere a more complex, organic shape. You will also build flippers out of primitive cylinders. These surfaces can then be bound to skeleton joints to begin setting up fluid character deformations when Salty is animated in Lesson 20.



Salty

In this lesson you will learn the following:

- How to create a new layer
- How to work with selection modes and selection masks
- How to model a character using CV edits
- How to build skeleton joints
- How to bind the character's skin to the skeleton
- How to develop a shading group for the seal skin

File management

In this lesson, you will be working on a single file. In order to manage the parts of this file, you will use layers to help organize your work into manageable parts.

1 Set project

A project directory is needed that contains sub-directories for different types of files that relate to your project.

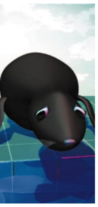
- Go to the **File** menu and select **Project → Set...**
- Open the *learningMaya* directory and then click on *projectFive*
- Click on the **Accept** button to make *projectFive* your current project.

  This existing project includes texture files to help you with your work.
- Select **File → New Scene**.

2 Create a new layer

To keep all of the parts of Salty the seal together, you will use a layer dedicated to this character.

- Select **Window → Layer Editor...**
- Press **New Layer**
- **Double-click** on *Layer1* and rename it *Salty*.
- Click **OK**.



Building Salty's body

The first step in building Salty, is to build the body and head. You will create these out of a single surface by sculpting a primitive sphere. To create the sphere, you will edit its input node using the show manipulator tool.

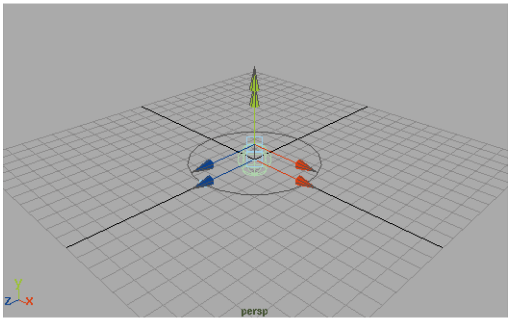
1 Place a primitive sphere

- Select **Create → NURBS Primitives → Sphere**.

2 Change the sphere's axis orientation

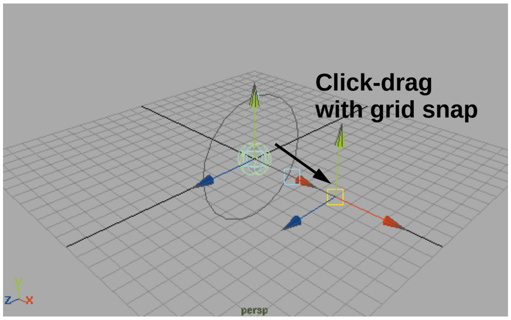
- In the Channel box, select the *makeNurbSphere1* input node.
- Select the **Show Manipulator** tool.

  Several manipulators appear that allow you to edit parts of the input node. In the case of the sphere, the manipulators control the position of the main axis of the sphere.



Sphere manipulators

- Press the **x** key to temporarily turn on grid snap and click-drag on the center of the top manipulator.
- Drag this manipulator down to the ground plane and place it along the X-axis with the help of grid snap.



Redefined axis

Now the sphere's polar axis is pointing along the X-axis. This is the direction in which you are going to build Salty.

Note: You could have also rotated the sphere to realign the polar axis. By using the input node, you keep the rotate values in the transform node at 0 0 0 which may be helpful when animating later.

3 Edit, then delete construction history

- In the Channel box, go to the *makeNurbSphere1* input node section and set the following:

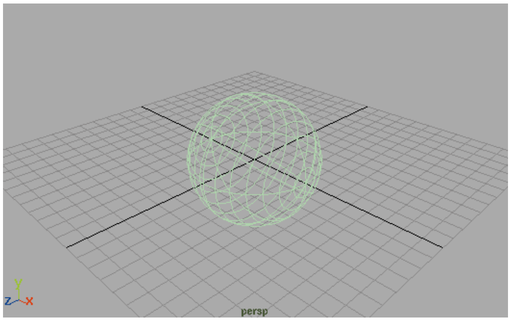
  Radius to **4**;

  Spans to **14**.

  This increases the size of the sphere and adds more isoparm spans. This will offer more flexibility when sculpting the surface into the shape of a seal.

- Select **Edit** → **Delete by Type** → **History.**

This deletes the sphere's input node. Since you are going to now pull CVs (control vertices) on the surface, the input node is no longer required.



Sphere

Note: Changing input node values after CV pulling is not recommended. Any changes to the number of spans later will negatively impact how your CV edits affect the surface.

Editing CVs

The sphere doesn't look very much like a seal at this point. You are now going to edit the positions of the surface's CVs to reshape the body.

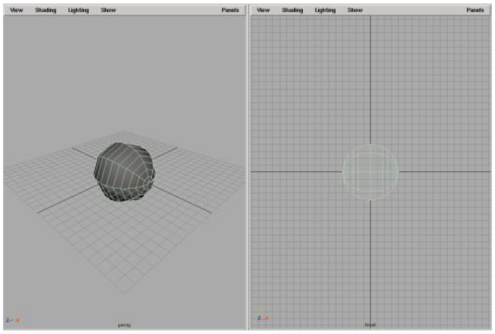
1 Set up your view panels

While you can pull CVs using any view panel combination, it is a good idea to view the model from all four view types – Top, Front, Side and Perspective.

- Select **Panels** → **Layouts** → **Four**.

Tip: Remember as you work, you can press the spacebar quickly to pop any of these panels to full screen and back for more detailed edits.

- Make the Perspective view the active panel and press the **5** key to turn on hardware shading.

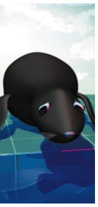

View Shading Lighting Show Panels
View Shading Lighting Show Panels
persp
front

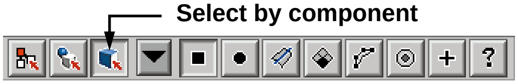
Four panel layout

2 Shape the sphere

In order to begin working with the CVs, you need to be able to select them.

- Pick the **Select** tool. Remember that you can use the **q** key.
- From the status line bar, click on the **Select by component** icon.

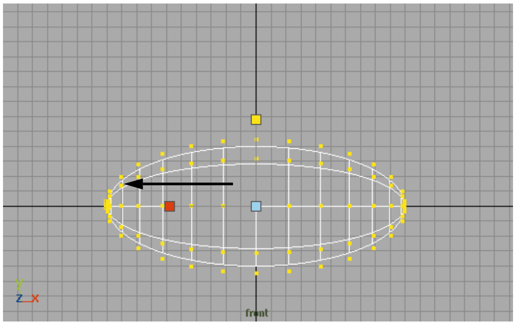




Select component mask

The sphere's surface is now highlighted in blue and its CVs are visible.

- Click-drag a selection box around all of the CVs
- Select the **Scale** tool.
- Click-drag on the X-axis handle to scale the sphere into the following shape:

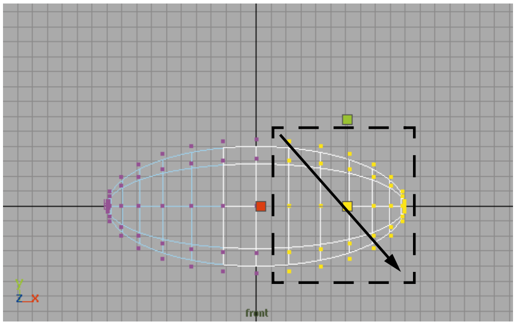


Scaled shape

3 Edit the CVs on Salty's head

When you first select a group of CVs, a special pivot is placed at the center of the selection. In some cases, you may want to move that pivot location.

- In the Front view, click-drag a selection box around the first few rows of CVs at the right end of the sphere.



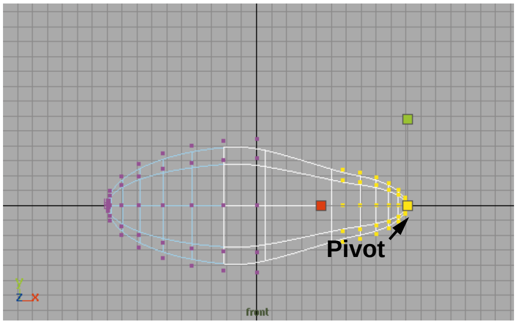
Selected CVs

The scale pivot is located at the center of the current selection.

- Press the **Insert** key to make the pivot manipulator visible.
- Drag the pivot to the right edge of the surface.

  This point will now act as the reference point for the scaling.
- Press the **Insert** key to return to the scale manipulator.
- Click-drag on the center scale handle to scale the head down to a smaller size.

  By dragging on the center of the manipulator, you make sure that you scale the CVs in all three axes.

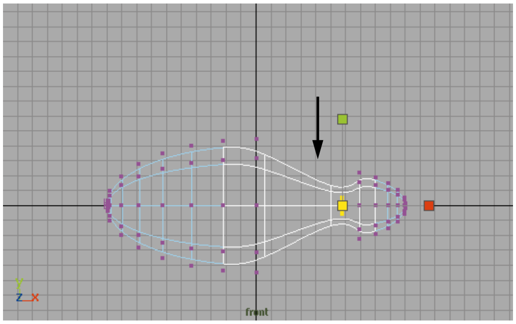


Scaled CVs

4 Edit the hulls on Salty’s neck

- Click-drag a selection box around the sixth row of CVs.
- Click-drag on the center manipulator handle to scale this row down.

  This time you don’t need to reposition the pivot because the center of the selection is the desired scale point.

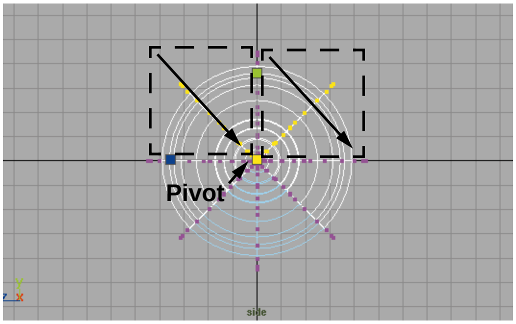


Neck CVs

This creates Salty’s neck.

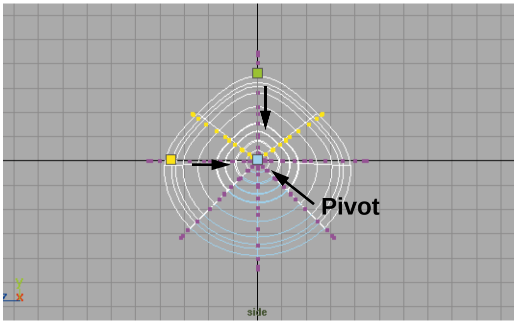
5 Taper Salty’s body

- In the side view panel, dolly into the view to see all of the shape.
- Click-drag a selection box around the CVs shown below. Use the **Shift** key to select both diagonal rows.
- Use the **Insert** key to move the pivot point of the selection to the origin.



Selected CVs

- Press **Insert** again to exit edit mode.
- Click-drag on the Y-axis scale handle to scale these points down.
- Click-drag on the Z-axis handle to scale these points in.

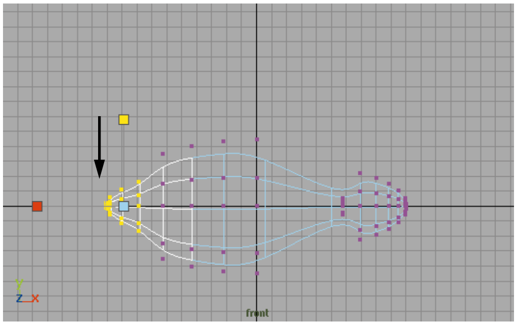


Scaled CVs

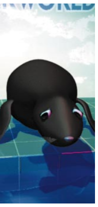
Note: In this case, you did not use the center scale handle since this would have also scaled along the X-axis. When working in one orthographic view, you must be aware of how your action may affect how the shape appears in another view.

6 Flatten the tail

- In the front view panel, **click-drag** a selection box around the last few rows of CVs.
- **Click-drag** on the Y-axis scale handle to scale these points down.



Tail CVs



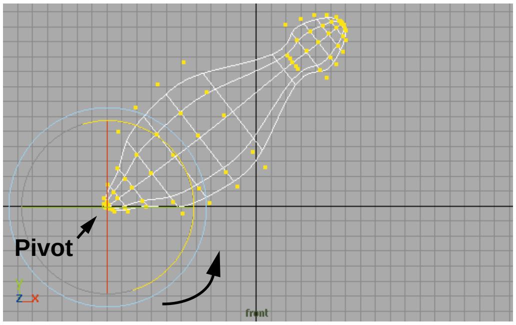
Positioning the CVs

You can further position the CVs by rotating them around the Z-axis. This will further define the shape of your character and begin to create a more seal-like shape.

1 Move all of the CVs

- **Select** all of the CVs.
- Select the **Rotate** tool.
- Use the **Insert** key to move the pivot point to the end of Salty's tail.
- Click-drag on the blue Z-axis handle to rotate the CVs up.

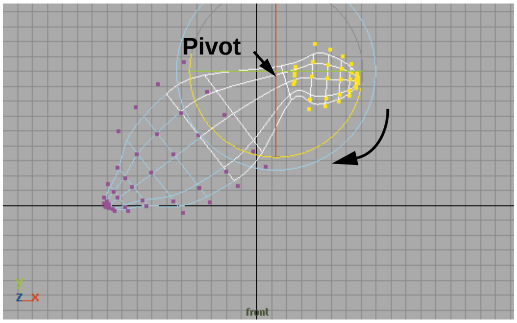
Check your other view panels to see if the rotation was successful.



Rotated CVs

2 Rotate the neck down

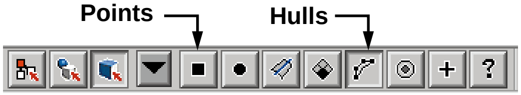
- **Select** all of the CVs on Salty's head.
- Use the **Insert** key to move the pivot point to the middle of Salty's neck.
- Click-drag on the blue Z-axis handle to rotate the head down.



Rotated CVs

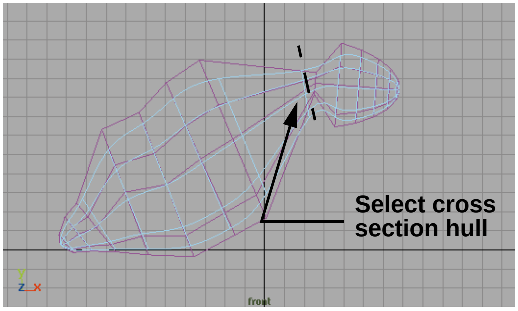
3 Adjust the surface hulls

- Click in empty space to unselect the current selection list.
- In the selection mask bar, turn the points selection mask off and turn on the hulls selection mask.



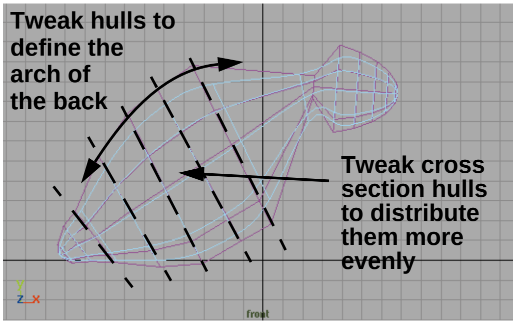
Selection masks

- Click on the cross section hull near Salty's neck. Be sure not to select any longitudinal hulls.
- **Rotate** the hull around the Z-axis.
- **Move** the hull forward as shown in the following:



Edited hull

- Use the **left/right arrow** keys to move between the cross section hulls.
- **Rotate**, **Move** and **Scale** the cross section hulls until they create a more even distribution while creating a nice arch along the back.


Tweak hulls to define the arch of the back
Tweak cross section hulls to distribute them more evenly
front

Rotated hulls

4 Save your work

- From the **File** menu, select **Save Scene As...**
- Enter the name *Salty1*.

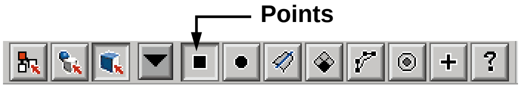
Finishing touches

The steps taken so far have been fairly mechanical. The result is a shape that resembles a seal, but lacks those extra details to really make it look right.

Pushing and pulling CVs is like sculpting a piece of clay and the best results come from those final tweaks. Now, you can use your own sculpting skills to finish Salty's torso.

1 Tweak the CVs

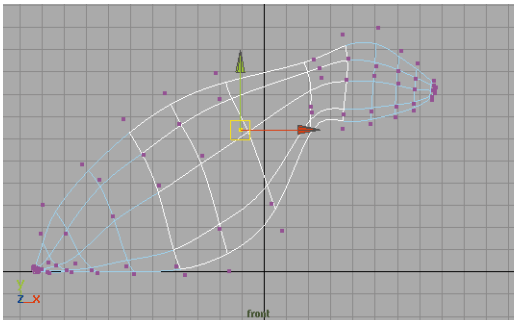
- Turn off the hull selection mask and turn on the points selection mask.



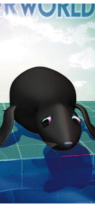
Selection masks

- In the Front view panel, **Select** and **Move** CVs to refine the shape of the character.

  Try to adjust the various isoparms so that they are relatively perpendicular to the center axis of Salty's body. Also make sure that you are selecting CVs that are on both sides of Salty to keep your edits symmetrical.



Tweaked CVs



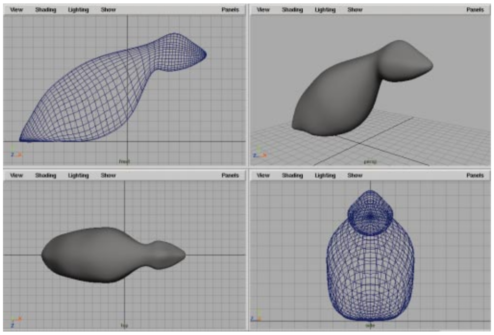
Tip: If you don't like an edit that you make, remember that you can **Undo** and **Redo** your changes.

2 Increase the surface smoothness

- Press **F8** to go back to **Select by Object** mode.

  Press the **3** key to display the surface with a higher smoothness setting to evaluate the surface.

  Press the **1** key when you want to go back to a low smoothness for modeling.



Shaded view of surface

Building skeleton joints

To animate Salty, you need to be able to deform the surface in a realistic manner. Moving and keyframing CVs would be a possibility, but would be a very time consuming process.

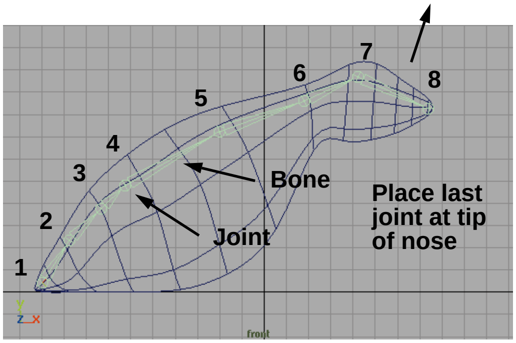
Skeleton joints can be used to create a structure that can be bound to your surface for more controlled deformations.

1 Draw the joints

The joints can be placed to suit the existing surface's shape.

- Go to the **Animation** menu set.
- Select **Display** → **Joint Size** → **25%.**
- Select **skeleton** → **Joint Tool**.
- Along Salty's spine and head, click eight times to place joints as shown in the image below.
- Press **Enter** to finish drawing the skeleton.

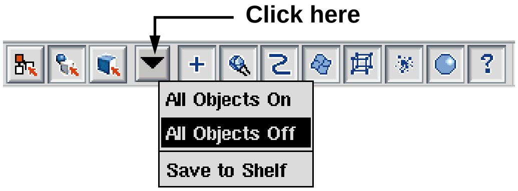
The joints are drawn as a series of small sphere's connected by the skeleton's *bones*.



Skeleton joints

2 Change your selection mask

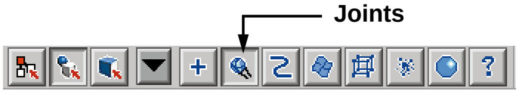
- From the black arrow next to the selection mode icons, select **All Objects Off**.



All Objects On
All Objects Off
Save to Shelf

Selection mask pop-up

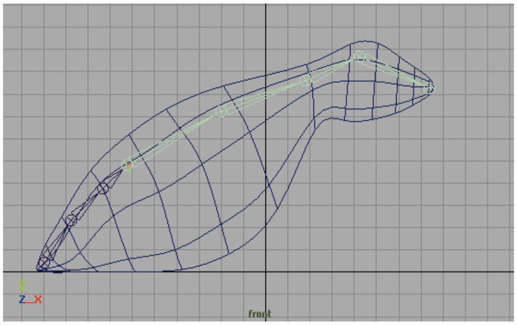
- Click on the **Select Joints** icon to activate this selection mask.



Select joints mask on

- Select the fourth joint up along the skeleton.

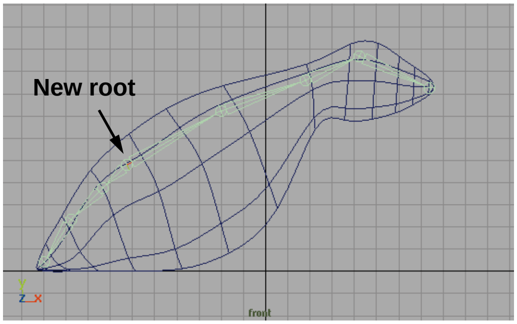
Tip: You can select a joint by either clicking exactly on the joint icon or by click-dragging over the next bone.



Selected joint

3 Reroot the skeleton

- Select **Skeleton** → **Reroot Skeleton**.



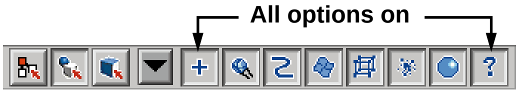
Rerooted skeleton

Bind the surface to the joints

1 Bind the skin

You must now bind the surface to the joints. Before you start, you need to change your selection masks to include surfaces.

- From the black arrow next to the selection mode icons, select **All Objects On**.



All masks on

Tip: It is very easy to forget that you have set a particular selection mask. When you try to select a different object type, you are unable to pick anything.

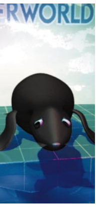
The joints should already be selected.

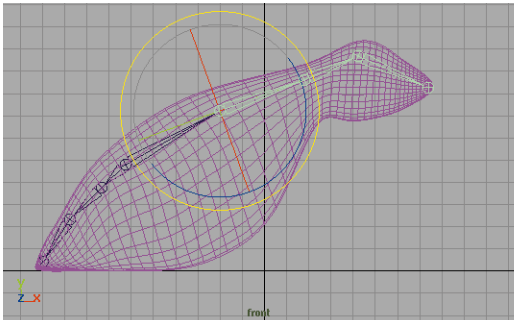
- Press the **Shift** key and click on the surface to add it to the selection.
- Select **Skin** → **Bind Skin** → **Rigid Bind.**

2 Test the results

- Select the surface and press the **3** key to increase the surface smoothness.
- Select the first joint past the root.
- Select the **Rotate** tool.

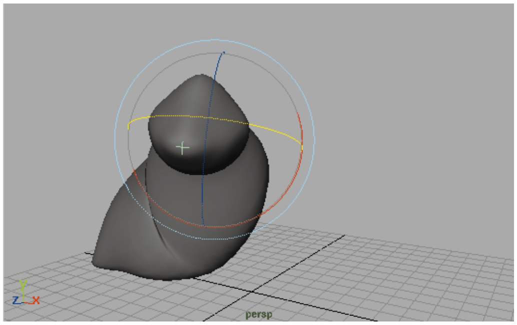
When you rotate a joint, you can see how the surface is set up to bend and twist with the joints.





Rotating the neck joint

- In the Perspective view, click-drag the rotate manipulator to see how the bound skin moves with the joints.



Testing the movement

3 Return to the bind pose

- Select the skeleton.
- Select **Skin** → **Go to Bind Pose.**

This sets the skeleton back to its bind pose.

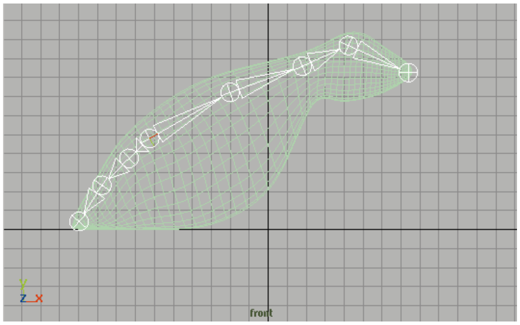
4 Save your work

Templating objects

To build the next body part, you need the main surfaces visible, but you don’t want them to interfere with your modeling. By templating the existing surface, you can use it for reference without fear of selecting it.

1 Select the surface

- Pick the **Select** tool.
- Select the surface and the bones.

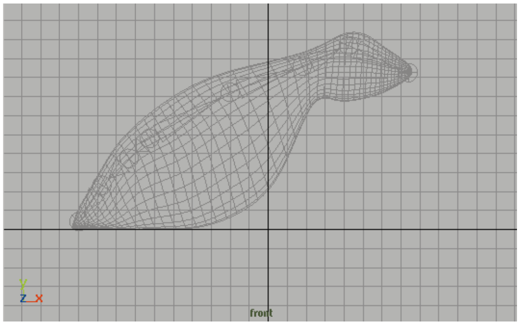


Selected pieces

2 Template the surface

- Assign the object to the *Salty* layer.
- Template the *Salty* Layer.

  Now these objects are visible but cannot be selected using the object selection mode.



Templated objects

- Click on the *Default* layer to make it the active layer.

Building the front flipper

To build a flipper, you will now sculpt a primitive cylinder. The techniques are similar to manipulating the sphere except that now you have some extra details to model.

1 Place a primitive cylinder

- Select **Create** → **Nurbs Primitives** → **Cylinder**.

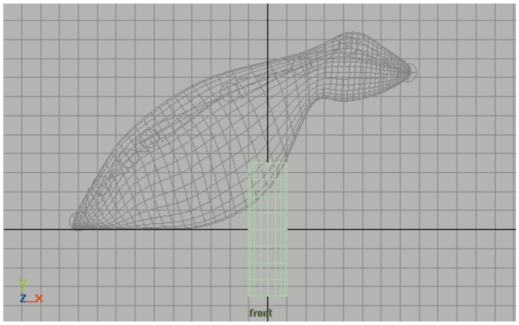
2 Edit, then delete construction history

- Select the *makeNurbCylinder* input node in the Channel box.
- Set the following:

  Sections to **16**;

  Spans to **8**;

  Height Ratio to **7**.



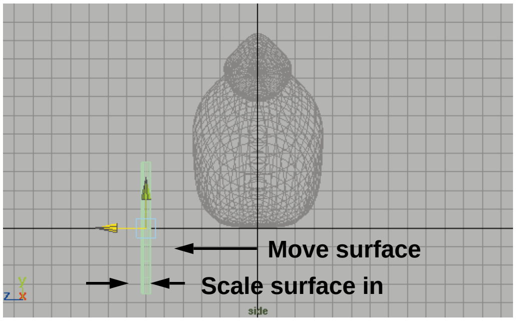
New cylinder

- Select **Edit** → **Delete by Type** → **History.**

3 Flatten and move the cylinder

- Select the **Move** tool.
- Click-drag along the Z-axis to place the cylinder just outside Salty's body.
- Select the **Scale** tool.

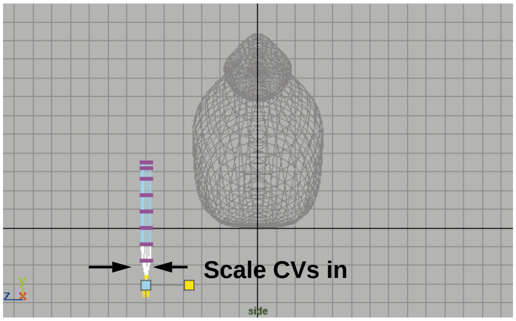
- In the Side view, click-drag on the blue Z-axis handle to scale the cylinder by a factor of about 0.25.



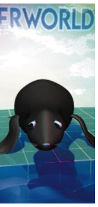
Flattened cylinder

4 Flatten the end of the cylinder

- Press **F8** to go into component select mode.
- Click-drag a selection box around the last three rows of CVs.
- Select the **Scale** tool.
- Click-drag along the Z-axis to squeeze these CVs until the end of the cylinder is flat.



Squeezed CVs

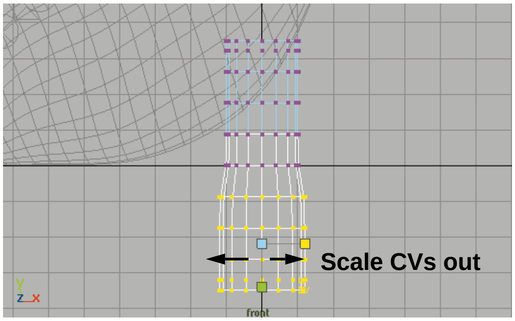


The side shape

From the Front view panel, you can see the side of the flipper. You will now reshape this profile to make the cylinder resemble a flipper.

1 Move some CVs to add shape

- In the Front view, dolly until the cylinder fills the panel.
- Press the **Shift** key and with the left mouse button add the next two rows to the selection.
- Click-drag along the X-axis to scale the CVs out a little.

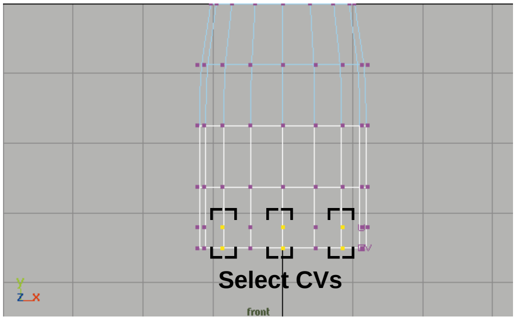


Edited CVs

2 Select the lower CVs

- Dolly in closer to the bottom five rows of CVs.
- Pick the **Select** tool.
- Click in empty space to deselect the current CVs.
- **Click-drag** a selection box around the lower middle two CVs as shown.

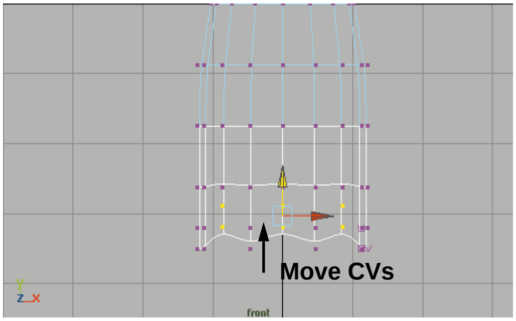
  You are actually selecting four CVs. Two in the front of the cylinder and two in the back.



Selected CVs

3 Move the CVs

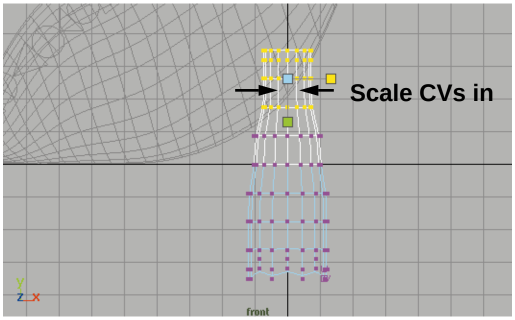
- Select the **Move** tool.
- Click-drag along the Y-axis to move the CVs up a little.



Edited CVs

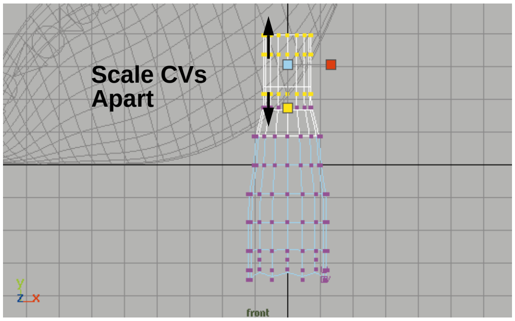
4 Scale the top CVs down

- Dolly out to see the whole flipper.
- Click-drag a selection box around the top four rows of CVs
- Select the **Scale** tool.
- Click-drag along the X-axis to squeeze these CVs to tighten the top of the flipper.



Scaled CVs

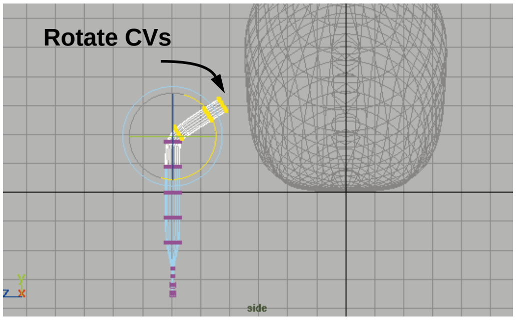
- Click-drag a selection box around the top three rows of CVs.
- Click-drag on the Y-scale handle to scale these CVs apart.



Scaled CVs

5 Rotate the top CVs down

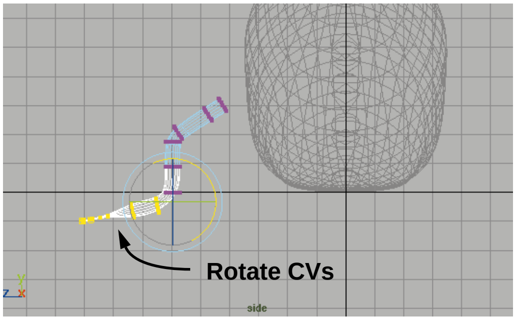
- Select the **Rotate** tool.
- Press the **Insert** key and in the Side view, move the pivot to a point in between the third and fourth rows of CVs.
- Again press the **Insert** key and, in the Side view, rotate the selection around the X-axis as shown in the following:



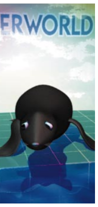
Rotated CVs

6 Rotate the bottom CVs up

- Select the bottom five rows of CVs.
- Press the **Insert** key and in the Side view, move the pivot to a point in between the fourth and fifth rows of CVs.
- Again press the **Insert** key and, in the Side view, **Rotate** the selection around the X-axis as shown below:



Rotated CVs

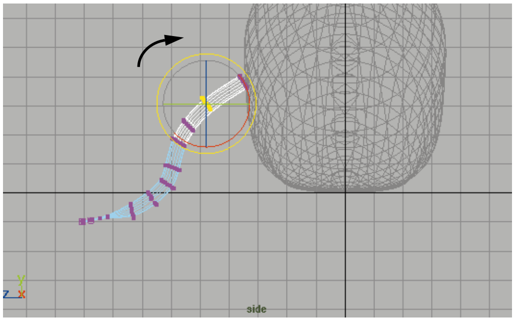


Refining the flipper

The rotations on the flipper are a little harsh. To improve the look of this profile, you can rotate and move the various rows of CVs to complete the shape.

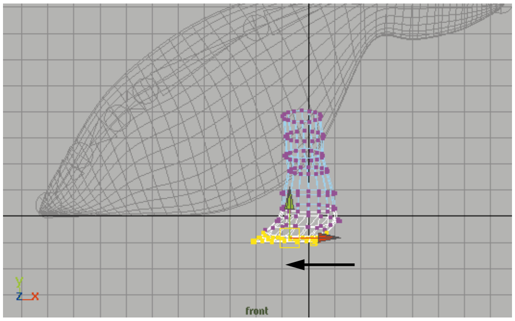
1 Adjust the rows of CVs in the Side view

- Select the rows of CVs on the flipper one at a time.
- **Move** and **Rotate** them to create a more sculpted shape:



Refined CVs

Be sure to consider how the flipper is looking in other views such as the Front view or the Perspective view.



Perspective view

2 Untemplate the body surface

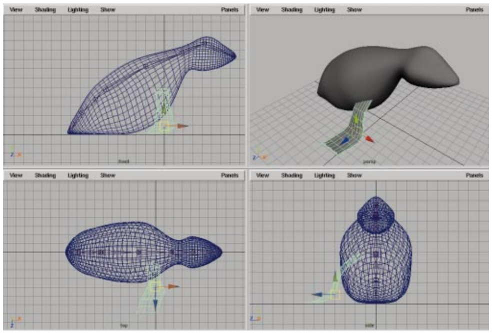
- Untemplate the *Salty* Layer.

Now, these parts are available for you to work with.

3 Move the flipper up to the body surface

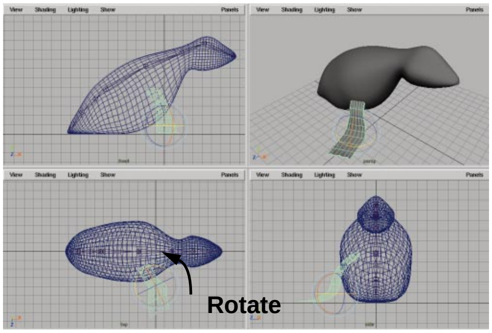
- Press **F8** to go to the **Object** select mode.
- Select the flipper surface.
- **Move** the flipper until its end is penetrating the side of the seal.

  Use all four views to confirm the placement.



Four views of Salty

- **Rotate** the flipper around the Y-axis to angle it a little as shown in the following:



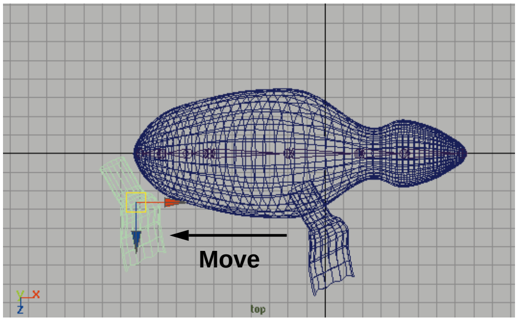
Rotate flipper

The back flipper

The back flipper will be created by editing a duplicate of the front flipper. It is useful to re-use pieces of geometry to make your sculpting easier.

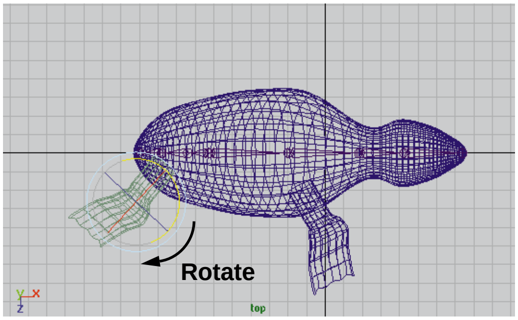
1 Build a back flipper

- Select the front flipper.
- Select **Edit** → **Duplicate**.
- Move the copy to the back of Salty.



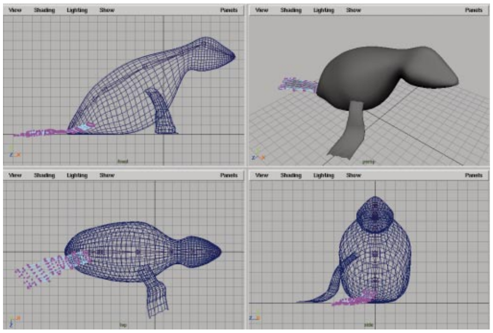
Back flipper

- **Rotate** the flipper around the Y-axis to line it up with the body.



Rotated flipper

- Edit the CVs and position the flipper at the back of Salty.



Back flipper

Mirroring the flippers

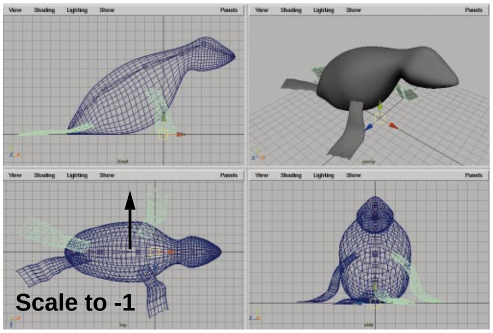
Rather than build flippers for Salty's left side from scratch, you can duplicate and mirror the existing flippers. A mirror can be achieved by scaling with a factor of -1.

1 Group, then duplicate the flippers

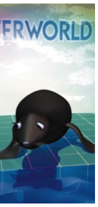
- Select the two flippers.
- Select **Edit** → **Group**.
- Select **Edit** → **Duplicate**.

2 Scale to -1

- In the Channel box, enter -1 in the *Scale Z* channel.



Mirrored flippers



3 Save your work

- Select **File** → **Save Scene**.

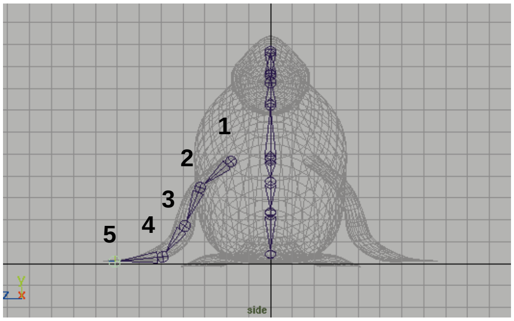
Add joints for the flippers

To connect the flippers to the rest of the body, you will build skeleton joints and then bind in the new surfaces.

1 Draw joints for the front flipper

- Select the front and back flippers and assign them to the *salty* Layer.
- Select the bones. From the *default* layer's menu, select **Assign selected**.
- **Template** the *Salty* Layer.

  Now you have templated the surfaces without the bones.
- Go to the **Animation** menu set.
- Select **Skeleton** → **Joint Tool**.
- In the Side view along Salty's front right flipper, click five times to place joints as shown in the following:



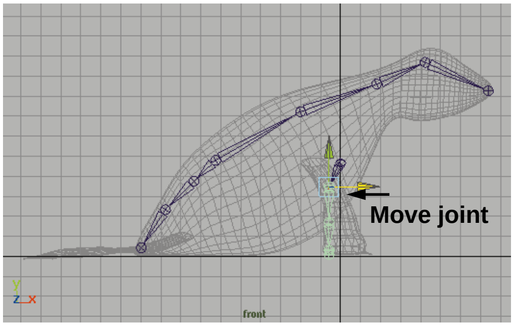
New joints

2 Move the third joint into place

- Select the second joint in the skeleton hierarchy.
- Select the **Move** tool.

- In the Front view, move the joint along the XY plane to the middle of the flipper.

  Oops! The joint and all the subsequent joints are moved at the same time. You don't want this.

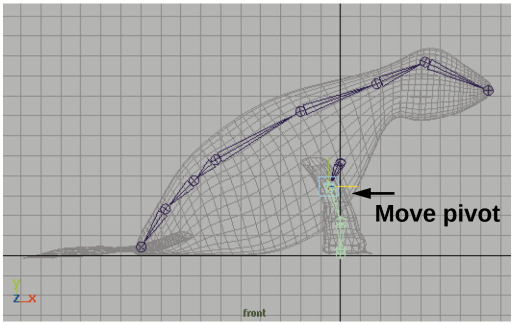


Moving the third joint

3 Set the pivot location for the joint

- **Undo** the last move.
- Press the **Insert** key to invoke the joint's pivot location.
- Move the pivot to the middle of the flipper.

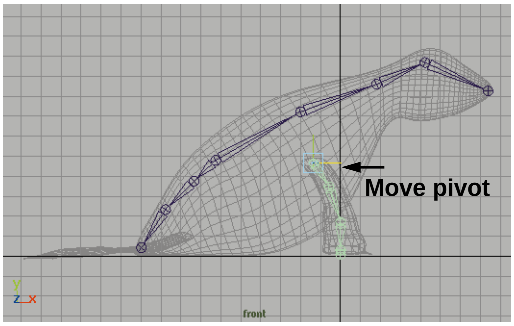
  Now this joint moves independent of the other joints. You can either move the joint or set the joint's pivot to relocate joints.



Moving the fourth joint's pivot

4 Edit the remaining joints

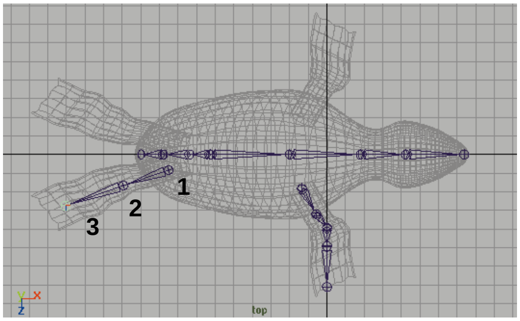
- Press the **up and down arrow** key to move the selection to the next joint.
- **Move** this joint's pivot into place.
- Press the **Insert** key to go back to the move manipulator.



All the joints in place

5 Add joints for the back flipper

- Select **Skeleton → Joint Tool**.
- In the Top view, draw three joints for the back flipper.
- Press **Enter** to complete the joints.



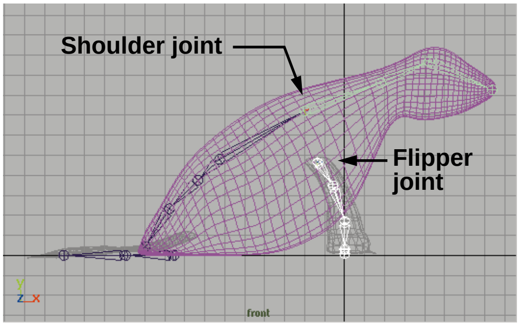
Back flippers

Joining the flippers to the body

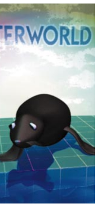
To make sure that the new joints work with the existing skeleton, you must connect them together. The logical place to join would be the hips and the shoulders.

1 Select the joints

- Using the Select Tool, pick the root joint of the front flipper.
- Press the **Shift** key and select the shoulder joint of the main spine.



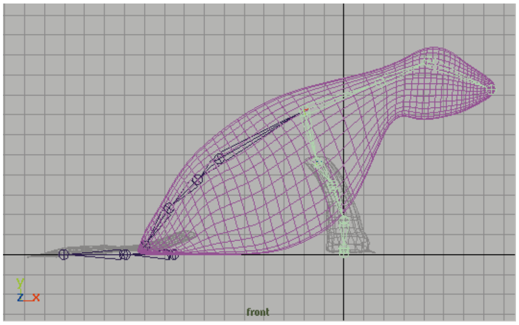
Selected joints



2 Connect the joints

- Select **Skeleton → Connect Joint** - ❐. In the Tool options, change the **Mode** to **Parent Joint**.

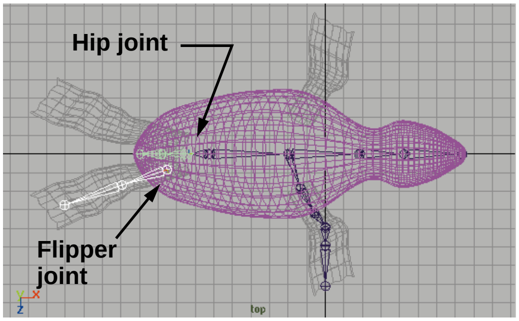
  This will place a bone between the two selected joints rather than moving the flipper joint up to the shoulder.
- Click **Connect** and then **Close**.



Connected joints

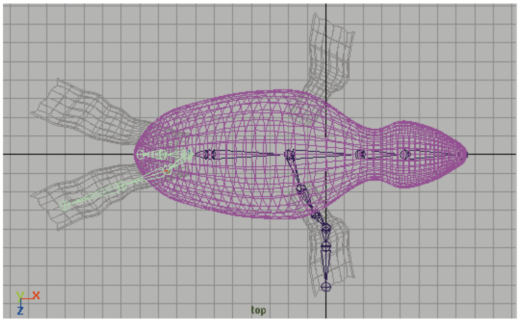
3 Connect the back joints

- Use the same technique outlined above to join the back flipper skeleton to the hip joint of the main skeleton.



Selected joints

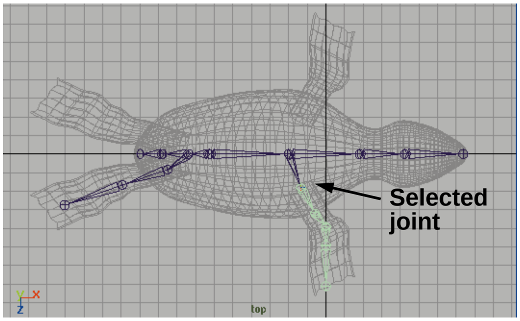
While you will not be animating the tail portion of Salty in this lesson, it is a good idea to complete the skeleton just in case.



Connected joints

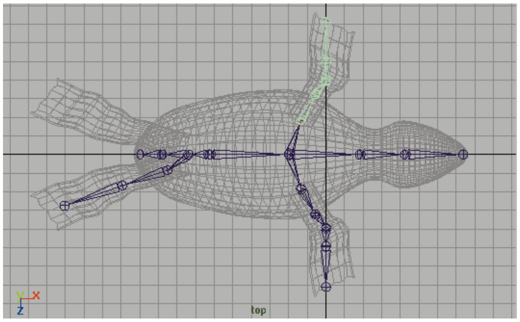
4 Mirror the joints

- Select the joint that used to be the root joint of the front flipper skeleton.



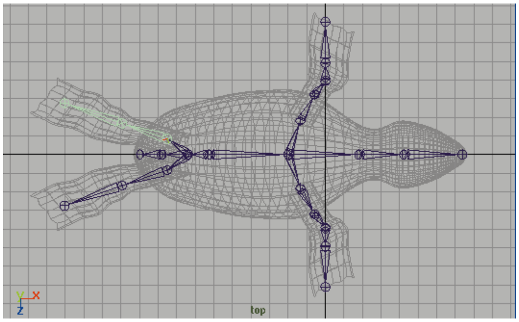
Selected joint

- Select **Skeleton → Mirror Joint.**



Mirrored joints

- Repeat for the back flipper.



Mirrored joints

Binding the surfaces

The flipper surfaces should deform with the skeleton, and therefore must be bound.

Binding a surface is a process where the surface's CVs are separated into sets which are then bound to the appropriate joints in the skeleton. When the joints move or rotate, the sets react as part of the joint hierarchy.

1 Untemplate the Salty layer

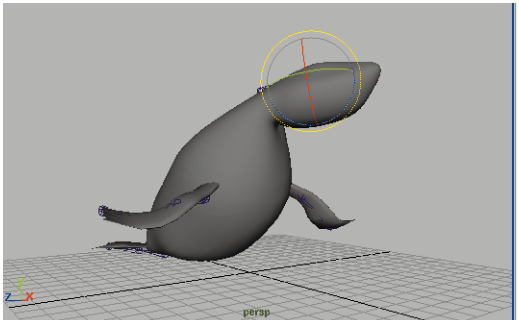
- Untemplate the *Salty* layer.

2 Bind all the surfaces

- **Select** the four flipper surfaces.
- Press the **Shift** key and add the main skeleton root to the selection.
- Select **Skin → Bind Skin → Rigid Bind.**

3 Test the motion

- **Rotate** various joints on the skeleton to test the resulting bind.



Rotated joints

4 Return to the bind pose

- Select the skeleton.
- Select **Skin → Go to Bind Pose.**

  This sets the skeleton back to its bind pose.

5 Assign all of the objects to a layer

- Select all of Salty's pieces and assign them to the *Salty* layer.

Salty's shading group

To define the look of Salty's skin, you will create a new shading group. The skin is to be dark and slick with a slight shinyness. You will set attributes on the material node to achieve this look.

1 Set your view panels

- Select **Panel → Saved Layouts → Hypershade/Outliner/Pers.**

2 Create a new shading group

You will create a shading group with a blinn material node which is suitable for shiny, metallic or wet surfaces.

- In the Hypershade, create a **Blinn** surface material.
- Rename the shading group and material nodes as follows:

  Shading Group node to *skinSG*;

  Material node to *skinBlinn*.

3 Set the common material attributes

The general color attributes are first set to create a basic dark color of the skin.

- Double-click on the *skinBlinn* node to open the Attribute Editor.
- Change the **Color** to a very dark brown by setting the following in the Color Chooser:

  Hue to **24**;

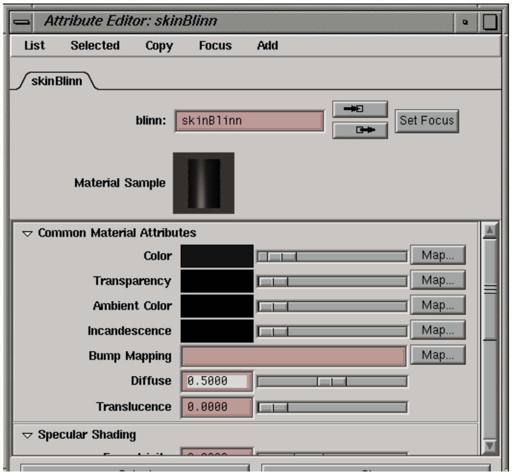
  Saturation to **0.3**;

  Value to **0.07**.

- In the Attribute Editor, set the following:

  Diffuse to **0.5**.

  Reducing the diffuse value decreases the amount of diffuse color that appears on the surface.



blinn: skinBlinn

Diffuse 0.5000

Translucence 0.0000

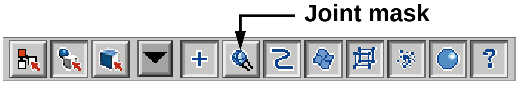
Color and diffuse settings

- Click on the **Minimize** button at the top of the Attribute Editor to minimize the editor.

  It is now readily available for use later.

4 Assign the shading group

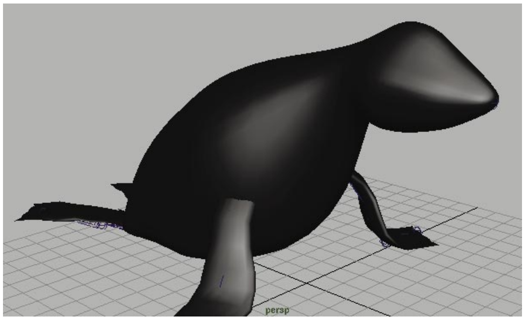
- Turn off the joint selection mask option. Now you will only pick surfaces.



Selection masks

- Assign the material to Salty's body, front flippers and back flippers.

  In the shaded view, you can see the material on the surfaces.



Shaded view of Salty

5 Set the specular shading attributes

The specular shading attributes define the way that the material appears in direct and ambient lighting conditions, creating highlights and hotspots on the surfaces. The seal's skin should appear soft, with a wet, reflective quality.

- Click on the Attribute Editor icon to maximize it.
- Click on the *skinBlinn* node in the Hypershade.
- Arrange this window so that you can see the shaded surfaces as you update the values.
- In the **Specular Shading** section, set the following:

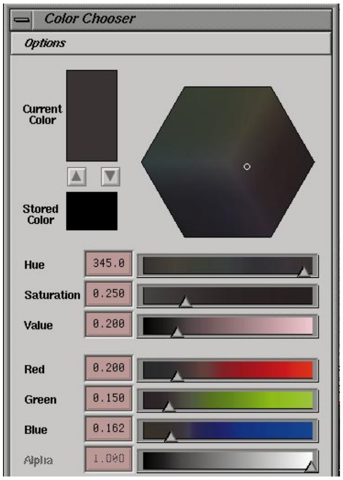
  Eccentricity to **0.6**.

  This creates larger areas of highlights which give a softer or more diffuse feel to the material.

  Specular Roll Off to **0.8**.

A higher value gives the impression of light being reflected at more oblique angles, making the surface appear more wet.

Specular Color to a darker value with a slight saturation of purple.



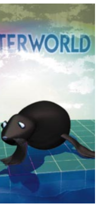
Hue 345.0
Saturation 0.250
Value 0.200
Red 0.200
Green 0.150
Blue 0.162
Alpha 1.000

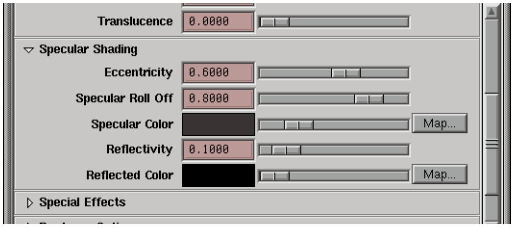
Specular shading attributes

In this book, HSV values of **255**, **0.25**, **0.2** were used. These settings darken the highlight and change its color from the warmer base color, to create a more complex surface quality.

Reflectivity to less than **0.1**.

The skin of the seal should only slightly reflect the surrounding surfaces.





Translucence 0.0000
Specular Shading
Eccentricity 0.6000
Specular Roll Off 0.8000
Specular Color
Reflectivity 0.1000
Reflected Color
Special Effects

Specular shading attributes

The various attributes of the material node allow you to clearly define how a surface looks. As you become more familiar with the available settings, you will be able to create shading groups that clearly represent real life materials.

6 Save your work

- Select **File** → **Save Scene**.

You will be using this scene file as the starting point for the next lesson.

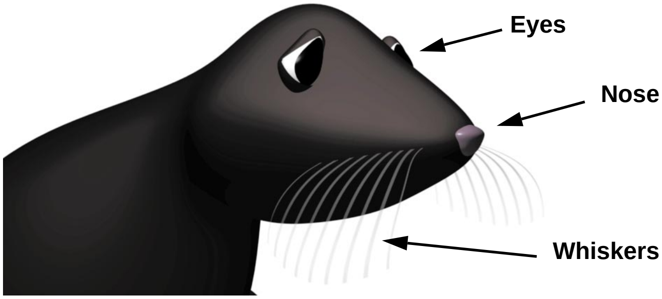
Conclusion

You are now off to a great start building Salty. In the next lesson, you will add facial details with special attention to the eyes which will be set up for animation later.

19 Adding Facial Details

In this lesson, you will develop Salty's facial features. You will start by building an eye assembly that will allow you to control where Salty is looking. You will also create a control object with special attributes that will let you set where the eyes look, how the eyes blink, and how the shape of the eyes express Salty's feelings. The goal is to explore how to model using Maya's procedural capabilities to help make it easier to animate all aspects of the eyes in later lessons.

In addition, you will add whiskers and a nose. All of these elements will be texture-mapped to prepare them for rendering later on.



Salty's facial details

In this lesson you will learn the following:

- How to build an eyeball hierarchy
- How to use an aim constraint
- How to apply lattice and cluster deformers
- How to add an attribute to a node
- How to use Set Driven Key
- How to use transparency maps

Initial set-up

To start this lesson, will continue with your file from the last lesson as you continue to build Salty.

1 Change the panel layout

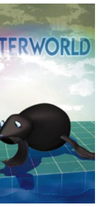
- Select **Panels → Saved Layouts → Four View**.
- Make the Perspective view panel active and press the **5** key to set smooth shading.

2 Template the existing objects

- In the Layer Bar, click with **RMB** over the Salty layer button and select **Template.**

Building Salty's right eyeball

To start building Salty's eye, you will build and texture an eyeball centered around the origin. Later, you will position it more closely to Salty's head.

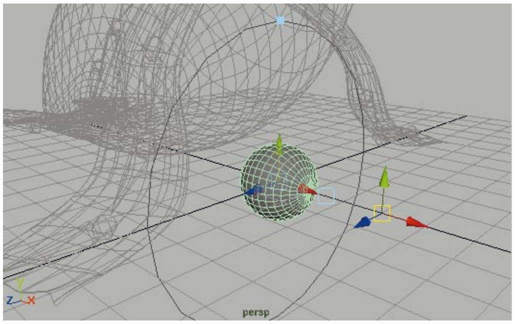


1 Place a primitive sphere

- Select **Create → NURBS Primitives → Sphere**.
- Press the **3** key to increase the display smoothness of the sphere.
- In the Channel box, rename the node to *eyeball.*

2 Use show manipulator tool to edit axis

- Dolly in to see the sphere from a closer point of view.
- In the Channel box, select the *makeNurbSphere* input node.
- Select the **Show Manipulator** tool.
- Press the **x** key to temporarily turn on grid snap and click-drag on the center of the top manipulator.
- Drag this manipulator down to the ground plane and place it along the X-axis with the help of grid snap.



New axis location

This orientation will facilitate the mapping of a ramp texture to the sphere later.

3 Edit the input node values

- In the Channel box, change the following attributes for the *makeNurbsSphere* node:

  Radius to **0.5**.

4 Create the eye shading group

The seal's eyes will use a phong material node that is shiny with a bit of translucence.

- In the Hypershade, Create a phong material node, and rename the shading group node to *eyeSG* and the material node to *eyePhong*.

5 Edit the material attributes

- Double-click on the *eyePhong* material node to open the Attribute Editor.

- Set the following attributes:

  Color to white;

  Diffuse value to **1.0**;

  Click on the color swatch next to **Incandescence** to set the value to **0.15**.

  The illuminated and shaded regions on the material sample are now quite bright.

  This will help make Salty's eyeball shine in any lighting.

6 Assign the shading group

- Assign the *eyeSG* node to the *eyeball* surface using the technique taught in earlier lessons.

7 Add a ramp color texture

To help differentiate the pupil from the white of the eyes, you will use a *procedural ramp texture*.

- In the Attribute Editor, click on the **Map** button next to **Color**.
- In the Create Render Node window, choose a **Ramp** texture.
- Make the Perspective view the active panel and press **6** to turn on hardware texturing.

Note: If you do not have hardware texturing, then you can continue with smooth shading on and then you can test the eyeball later with a test rendering.

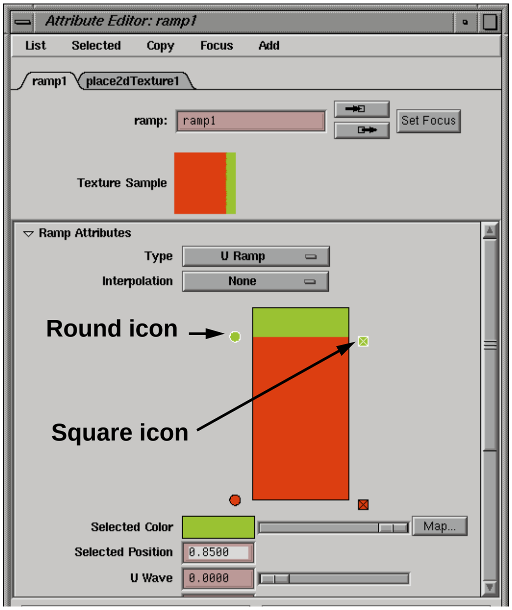
8 Edit the ramp

- In the Attribute Editor, change the ramp attributes as follows:

  Type to **U Ramp**;

  Interpolation to **None**.

  This creates a harsh break between the first two colors on the ramp.
- Click on the square icon to the right of the blue color indicator to delete this color.
- Click on the round icon to the left of the green color indicator to select it.
- Click-drag on the round icon to move the indicator up until its is set to around **0.85**.



ramp: ramp1

Type: U Ramp

Interpolation: None

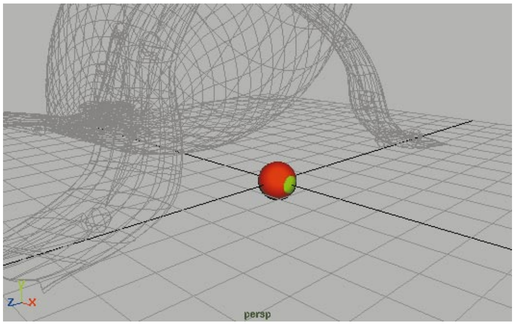
Selected Color

Selected Position: 0.8500

U Wave: 0.0000

Ramp setup

If you have hardware rendering, then you will see the texture update in the Perspective view.



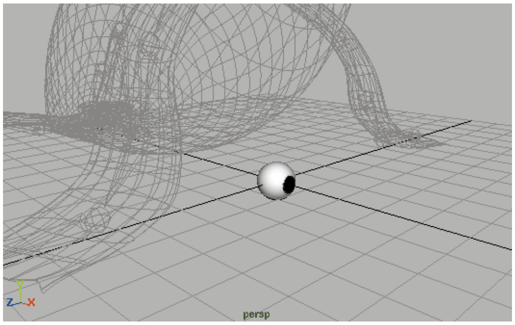
Eyeball with texture

9 Edit the ramp's colors

- In the Attribute Editor, drag the green color indicator's **Selected Color** slider all the way to the left.

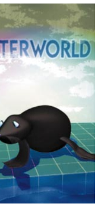
This makes this color turn to black.

- Click on the round icon for the red color indicator.
- Click on the **Selected Color** swatch to open the Color Chooser.
- Change this color to white.



Eyeball with updated texture

- Close or minimize the Hypershade and the Attribute Editor.

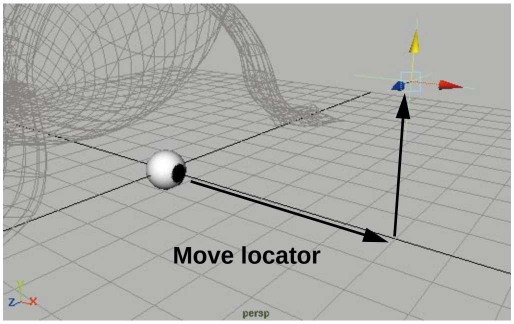


Creating a target for the eyeball

To animate the eyeball later, you will want to be able to control what it is looking at. You will first create a locator object, then constrain the eyeball to aim towards it.

1 Add and position a locator object

- Select **Create** → **Locator**.
- **Move** the locator along the X-axis to move it in front of the eyeball.
- **Move** the locator along the Y-axis a little to place it above the ground.



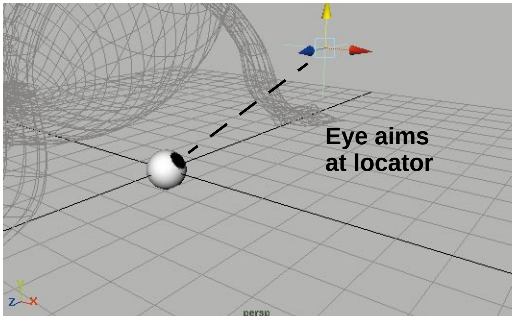
Locator object moves

2 Aim the eyeball at the locator

- Change to the **Animation** menu set.
- Press the **Shift** key and select the *eyeball* surface.

  Now the locator and the eyeball are selected. Since the eyeball was selected last, it is indicated in green. Since you are aiming the eye, it is important that it is picked last.
- Select **Constrain** → **Aim**.

  Now the eye is looking at the locator.
- **Move** the locator, on its own, in all axes to test the aiming of the eye.



Eyeball with texture

- In the Channel box, set the following for the *locator1* node:

  Translate X to **7**;

  Translate Y to **0**;

  Translate Z to **0**.

  Now the eyeball is looking straight ahead.

Creating the eyelid

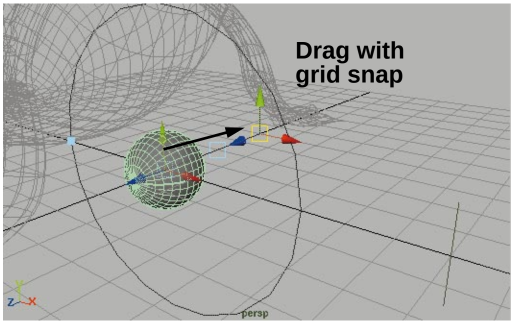
You are now going to build an eyelid out of another primitive sphere. The input node will be used to set the start and end sweep of the sphere which help create the blinking action.

1 Place a second primitive sphere

- Select **Create → NURBS Primitives → Sphere**.
- In the Channel box, rename the *nurbSphere* node to *eyelid.*
- Increase the object's smoothness using the techniques taught in earlier lessons.

2 Edit the sphere's axis

- In the Channel box, select the *makeNurbSphere* input node.
- Select the **Show Manipulator** tool.
- Press the **x** key to temporarily turn on grid snap and click-drag on the center of the top manipulator.
- Drag this manipulator down to the ground grid and place it along the Z-axis with the help of grid snap.



New sphere

3 Edit the input node attributes

The sphere's input node contains attributes that control the start and end sweep of the shape. You will set these now to look like a partially opened eye. After, you will animate these attributes to make Salty blink.

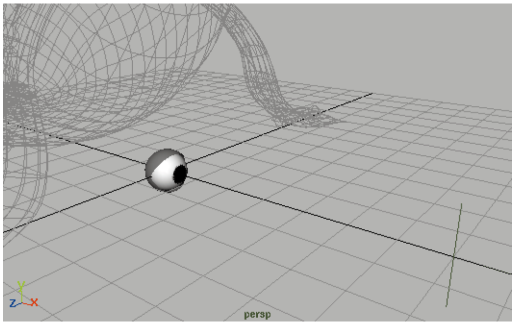
- In the Channel box, change the following attributes for the *makeNurbsSphere* node:

  Radius to **0.55**;

  Start Sweep to **60**;

  End Sweep to **300**.

  The two sweep attributes let you open and close the lid surface using the construction history built into the sphere's input node.



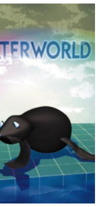
Edited eyelid

Note: When you were modeling the body surfaces, you deleted history. For the eye, you will keep history since the start and end sweep can be used to blink the eye. This will be an example of procedural animation as the keys are set on the input node to animate the sphere.

4 Assign the seal shading group

You can now assign the existing seal skin shading group to the eyelid surface.

- Change to the **Render** menu set.
- Select **Shading** → **Assign Shading Group** → **skinSG**.

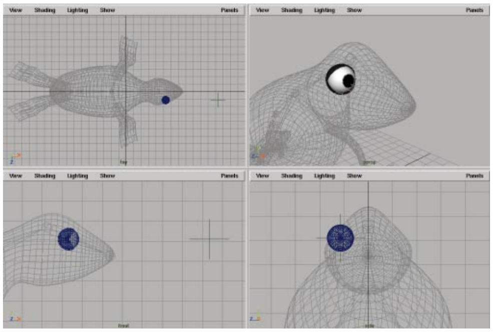


5 Group the eyelid and the eyeball

- Select both the *eyelid* and the *eyeball*.
- Select **Edit** → **Group**.
- In the Channel box, change the name of *group* to *rightEye*.

6 Move the parts of the eye

- Press the **Shift** key and select *locator1* to add it to the current selection list.
- **Move** these objects up and across to place the eye group on the right side of Salty's head.



Positioned objects

7 Save your work

- From the **File** menu, select **Save Scene As...**
- Enter the name *Salty2* next to the file's path.

The Hypergraph

Whenever you create a group of objects, you create a transform hierarchy. In earlier projects, you learned how to look at these hierarchies in the Outliner. The *Hypergraph* offers a more diagrammatic view of the same information.

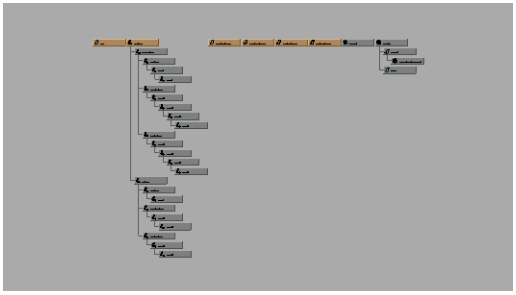
When working with hierarchies, you can either work with the top or *root* node, or you can work with the lower *leaf* nodes. Complex animations are possible when you combine animation at different hierarchy levels.

1 Open a Hypergraph panel

- Click in empty space to clear your selection.
- From the Top view panel, select **Panels** → **Hypergraph...**

- From the Hypergraph panel, select **View → Frame All**.

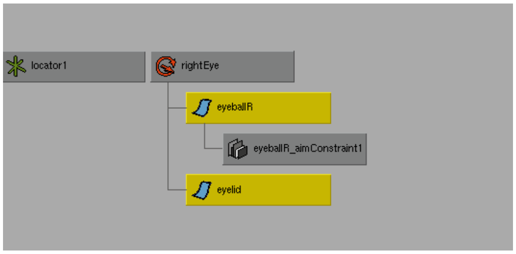
  This shows you all the transform node hierarchies that exist in the model now. This includes the skeleton joint hierarchy and the new eye hierarchy.



Hypergraph panel

2 Select the eye group

- Click-drag a selection box over the *rightEye* group.
- From the Hypergraph panel, select **View → Frame Selection**.
- Use the **Alt** key and the left and middle mouse buttons to dolly out of the scene. The same view tools apply here as in other panels.

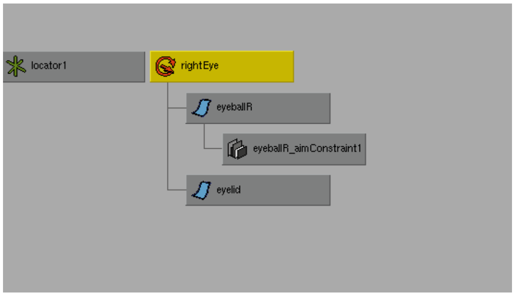


Hypergraph panel

In the Hypergraph, you can see that the two lower nodes in the hierarchy have been picked even though you may have wanted to pick the upper node.

When you are in object selection mode, you are always picking the lowest node in the hierarchy.

- Press the **up arrow** key to move up the hierarchy to the *rightEye* group node.



Hypergraph panel

Deforming the eye

The eye is looking good – but it needs more character. To give it a more dynamic shape, you can apply a *lattice deformer*.

Lattice deformers let you manipulate the shape of a surface using an external frame. The lattice frame has its own control points that can be edited just like CVs to let you deform both the eyeball and the eyelid as a single group.

1 Apply a lattice to the eye group

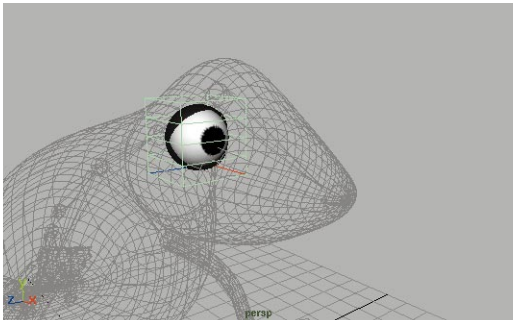
- Change to the **Animation** menu set.

  Select **Deform → Create Lattice - ❐**.

- In the Lattice Options window, set the following:

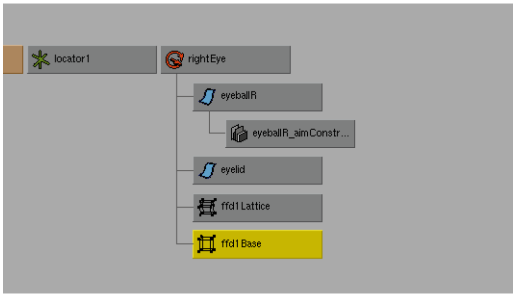
  Parenting to **On**.

- Click the Create button.



Eye lattice

In the Hypergraph, you can now see two new nodes parented under the *rightEye* group. By parenting these to the group, you can move them all together to maintain the deformations as Salty's head moves.



Hypergraph panel

2 Scale the lattice base object

The lattice base is a special node designed to contain the geometry that is to be deformed. If your objects move out of the base, then they will no longer be deformed.

To be on the safe side, you should scale this node out to make sure it encompasses the entire eye, even when it is deformed later.

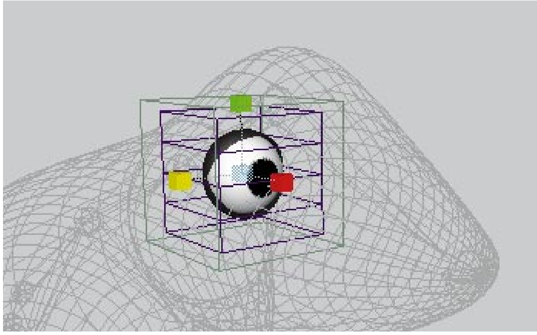
- In the Hypergraph panel, click on the *ffd1Base* and the *ffd1Lattice* nodes.
- In the Channel box, edit the following attributes:

  Scale X to **1.5**;

  Scale Y to **1.5**;

  Scale Z to **1.5**.

  This scales out the lattice and its base, allowing the eye to be encompassed more completely by the lattice. The eye group remains the same size since both parts were scaled.



Scaled lattice base

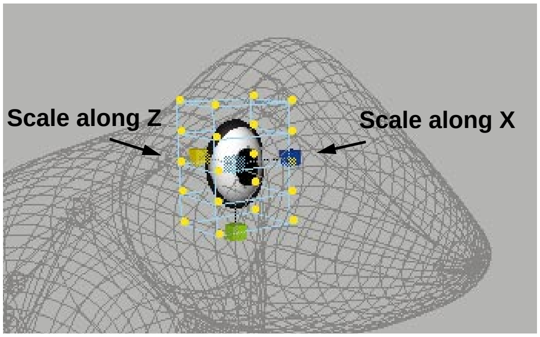
3 Scale all the lattice points

You will now work with the lattice node which lets you apply the actual surface deformations.

- In the Hypergraph panel, select only the *ffd1Lattice* node.
- Press the **F8** key to go into component mode.

  The lattice points are now visible and available for editing.

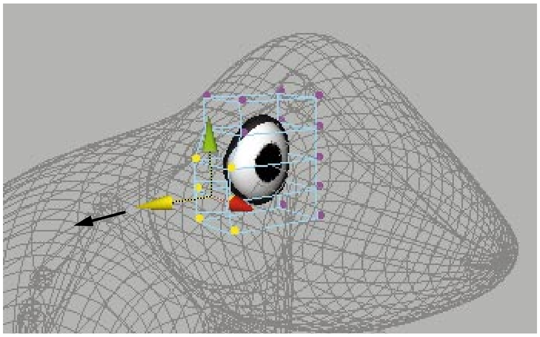
- Click-drag a selection box around all of the lattice points.
- **Scale** these points along the X and Z axes to scale the lattice points into the following shape:



Scaled lattice points

4 Move the bottom lattice points

- Click-drag a selection box around the six points on the lower side of the lattice.
- **Move** these points along the Z-axis handle to push the points out.



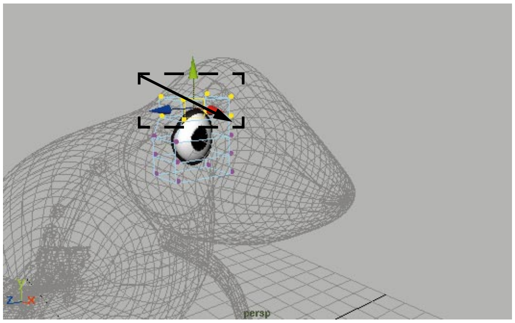
Pushed lattice points

Adding a cluster deformer

Another type of deformer is a *cluster*. Clusters let you create a set of CVs or lattice points and give these points their own special pivot point for specific scaling and rotational edits. This cluster will be used to make Salty's eyes appear to get worried.

1 Add a cluster to the top lattice points

- Click-drag a selection box around the top eight lattice points.

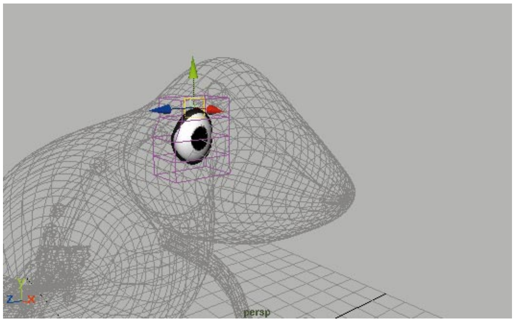


Selected Lattice points

- Select **Deform** → **Create Cluster**.

  A cluster is now created using these points. It is indicated using a small C icon.

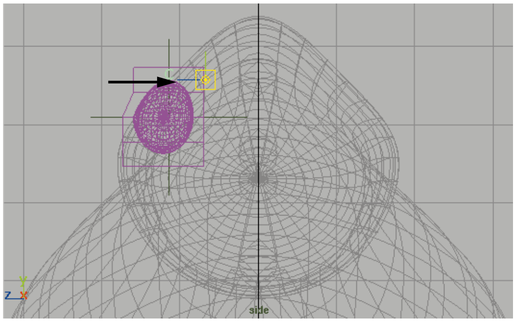
- In the Channel box rename the *clusterHandle* to *eyeCluster*.



New cluster object

2 Edit the pivot location of the cluster

- Press **F8** to go back to object selection mode.
- Press the **Insert** key.
- Click-drag the cluster's pivot along the Z-axis to the edge of the lattice.

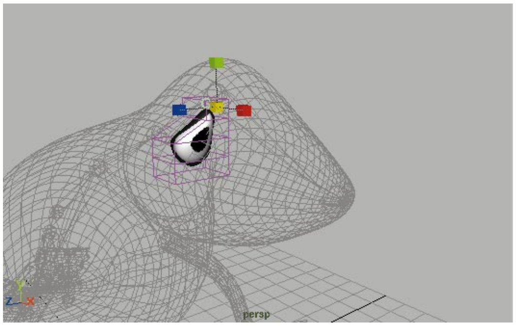


New pivot location

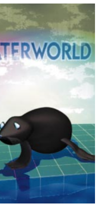
3 Scale the cluster

- Press the **Insert** key to get out of edit mode.
- **Scale** the cluster in along the Z-axis to a value of about **0.5**.

  This gives Salty a worried look.



Scaled cluster



Positioning the eye

When you positioned the eye earlier, you used the templated lines for reference. You will now make sure that the eye is not intersecting the main body surface.

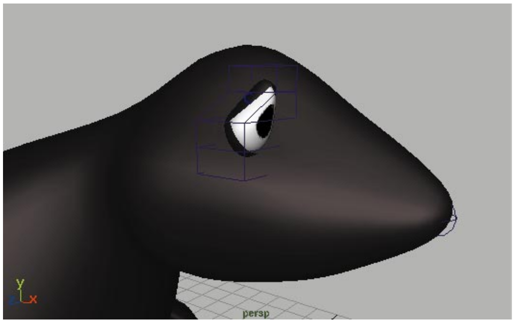
1 Untemplate the various surfaces

- Untemplate the *Salty* layer.

  The eye may not be positioned correctly in relation to the head surface causing intersecting surfaces. You should reposition it to create a more accurate relationship between these pieces.

2 Reposition the eye group

- In the Hypergraph, select the *rightEye* group node.
- **Move** the group until it is correctly positioned in relation to the main surface.



Repositioned eye group

Creating the second eye

You will create the second eye by duplicating the first. Duplicating the eye with all of its connections will make sure that the input nodes for the sphere and the constraint are included. This is important to ensure that both eyes work in the same manner.

1 Duplicate the whole hierarchy

- In the Hypergraph, select both the *rightEye* group node and the *locator1* node.
- Select **Edit** → **Duplicate** - ❐.
- In the Duplicate Options window, set the following:

  Duplicate Upstream Graph to **On**.
- Click on **Duplicate** and **Close**.

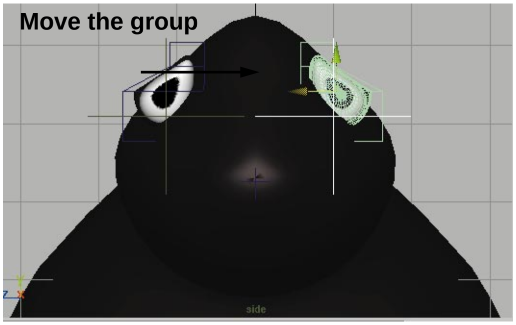
2 Mirror the new eye group

- In the Channel box, change the **Scale Z** value to **-1**.

  This mirrors all of the selected parts.

3 Move the new eye group

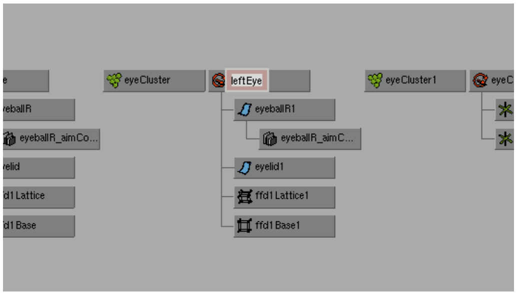
- **Move** the selection along the Z-axis until the new eye group sits on the opposite side of the face.



Both eyes in place

4 Rename the new eye group

- In the Hypergraph panel, press the **Ctrl** key and double-click on the *rightEye1* node.
- Type the name *leftEye* and press the **Enter** key.



Renaming the left eye group

Building the eye control node

You could now animate the eyes by keying the position of the locators to determine where Salty is looking, the start and end sweep of the *makeNurbSphere* nodes (to make Salty blink), or the scale of the Clusters (to give Salty a worried look).

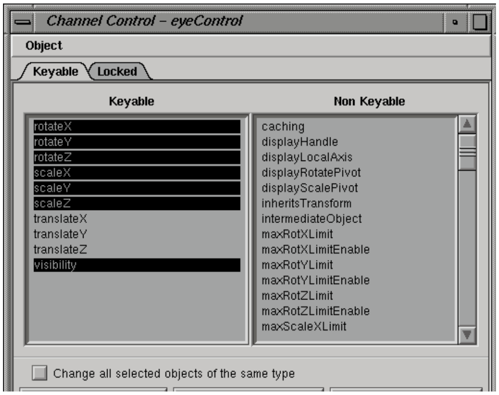
One problem, however, is that all of these parts are buried within the eye assembly. You will now group the two locators and create an eye control, node which will be linked using expressions to control all of the items mentioned above. By creating this control now, Salty will be much easier to animate in the next lesson.

1 Group the locators

- **Select** the two locator objects.
- Select **Edit** → **Group**.
- Select **Modify** → **Center Pivot...**
- Select **Display** → **Object Components** → **Selection Handles**.
- In the Channel box, rename the group node to *eyeControl*.

2 Restricting channels on the new group

- From the **Window** menu, select **General Editors → Channel Control...**
- In the Keyable list, press the **Ctrl** key and click on the attribute names as shown in the following:



Channel Control – eyeControl

Object

Keyable | Locked

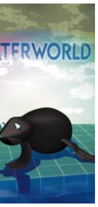
Keyable	Non Keyable
rotateX	caching
rotateY	displayHandle
rotateZ	displayLocalAxis
scaleX	displayRotatePivot
scaleY	displayScalePivot
scaleZ	inheritsTransform
translateX	intermediateObject
translateY	maxRotXLimit
translateZ	maxRotXLimitEnable
visibility	maxRotYLimit
	maxRotYLimitEnable
	maxRotZLimit
	maxRotZLimitEnable
	maxScaleXLimit

□ Change all selected objects of the same type

Channel Control window

- Press the **Move >>** button to move these into Non Keyable list.

  In the Channel box, you can now see only the three translate attributes.
- Close this window.



3 Add a worry attribute

So far, you have been working with attributes specific to different node types, and you have revealed hidden attributes. To control parts of Salty's eyes, you are going to add your own attribute that will be connected using expressions.

- Select **Modify → Add Attribute...**
- In the Add Attribute window, set the following:

  Attribute Name to *worry*;

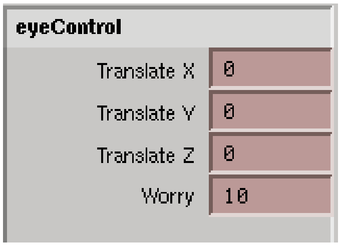
  Data Type to **Float**;

  Minimum Value to **0**;

  Maximum Value to **10**;

  Default Value to **10**.

  These settings create the value range for the attribute.
- Click the **Ok** button.



eyeControl

Translate X	0
Translate Y	0
Translate Z	0
Worry	10

New attribute in Channel box

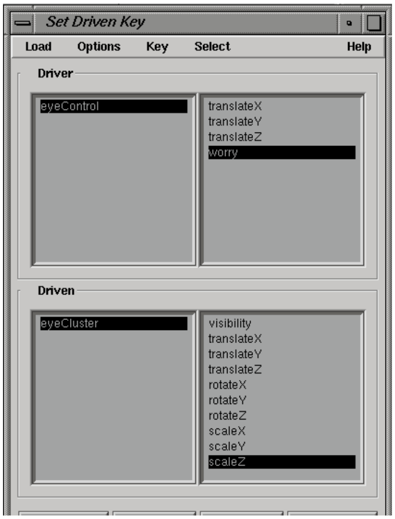
4 Use worry to drive the cluster

To drive the cluster's scaling along the Z-axis, you are going to use Set Driven Key. This tool lets you set up relationships between different attributes. You will set up the worry node to drive the scaling of the *eyeCluster*.

- Go to the **Animation** menu set.
- In the Hypergraph, select the *eyeCluster* node.
- Next, select **Animate → Set Driven Key → Set - ❐.**

  This opens a window that lets you link attributes. The *eyeCluster* node is loaded as the driven node.
- Click on the **scaleZ** attribute in the right-hand list.

- In the Perspective view, **Select** *eyeControl* using its selection handle.
- In the Set Driven Key window, click on the **Load Driver** button.
- Click on the **worry** attribute in the right-hand list.



Set Driven Key window

5 Key the scale positions

- In the Set Driven Key window, click on the **Key** button.

  Now, when the **worry** attribute is set to 10, the **scale Z** attribute will be set to 0.5.
- In the Channel box, change the **Worry** attribute to **0**.
- Select the *eyeCluster* node.
- In the Channel box, change the **Scale Z** attribute to **1**.
- In the Set Driven Key window, click on the **Key** button.

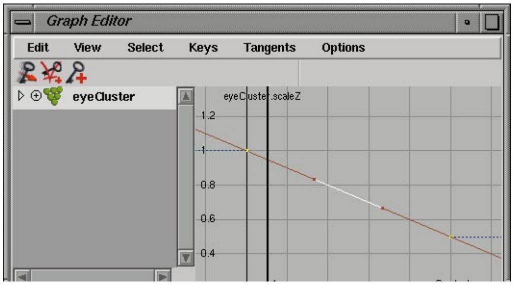
6 Test the results

- Select the *eyeControl* group using its selection handle.
- In the Channel box, click on the **Worry** channel name.
- In the Perspective view panel, click-drag with the middle mouse button to test the attribute.

  As you click-drag from 0 to 10, Salty's eye changes its expression.

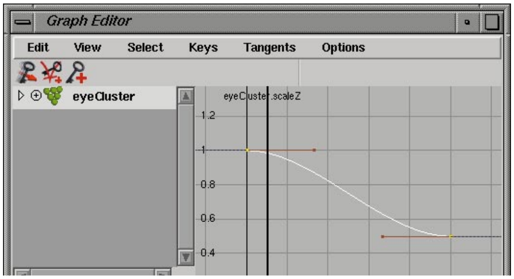
7 Edit the resulting graph curves

- Select the *eyeCluster* node.
- Select **Window → Animation Editors → Graph Editor...**
- From this window, select **View → Frame All**.
- Along one axis of the curve diagram is the cluster's *scale Z*, and along the other is the *eyeControl* node's **worry** attribute based on the keys set using Set Driven Key.



Set Driven Key anim curve

- Select the curve.
- Select **Tangents → Flat**.
- This changes the way in which the driven attribute is affected by the driver. You can edit this curve to define any relationship you desire.

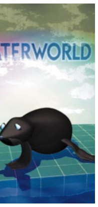


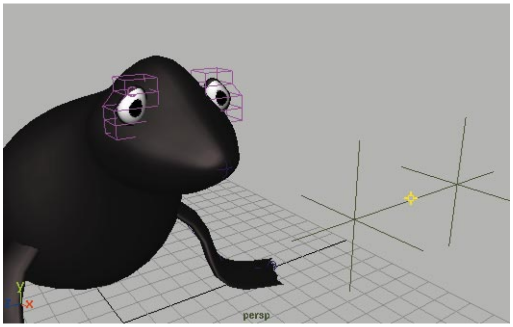
Edited anim curve

8 Repeat for the other cluster handle

- In the Hypergraph, select the *eyeCluster1* node.
- Use the Set Driven Key window to drive this node using the *eyeControl* node's **worry** attribute.
- When you are finished, set the *eyeControl* node's **worry** attribute to **7.**

  This creates a good neutral position. Later, you will animate this attribute to make Salty react to the bouncing of the ball in the next lesson.





Salty looking a little worried

9 Save your work

Adding a blink attribute

Another useful control would be a blink attribute. This would allow you to control the opening and closing of the eyes using a single attribute, instead of working with the two sweep attributes on two separate input nodes.

1 Add a new attribute

- Make sure that the *eyeControl* group is selected.
- Select **Modify** → **Add Attribute.**
- In the Add Attribute window, set the following:

  Attribute Name to *blink*;

  Data Type to **Float**;

  Minimum Value to **0**;

  Maximum Value to **10**;

  Default Value to **10**.
- Click on the **Ok** button.

2 Use blink to drive the sweep angles

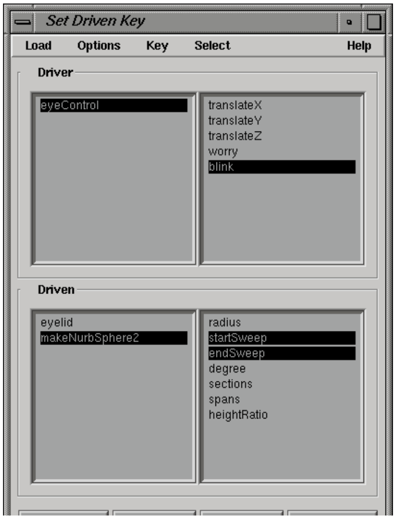
To drive the opening and closing of the *eyeLid*, you are going to again use Set Driven Key. This time you will use the **blink** attribute to drive the start and end sweep of the eyelid's *makeNurbSphere* input node.

- Go to the **Animation** menu set.
- In the Hypergraph, select the *eyelid* node within the *rightEye* hierarchy.
- In the Channel box, click on the *makeNurbSphere* input node.
- Select **Animate** → **Set Driven Key** → **Set - ❐.**

In the Set Driven Key window, you will see the *eyeLid* node loaded in the Driven section.

- Click on the **makeNurbSphere** node in the left-hand list, then click on both the **startSweep** and **endSweep** attributes in the right-hand list.
- In the Perspective view, **Select** *eyeControl* using its selection handle.
- In the Set Driven Key window, click on the **Load Driver** button.

  This updates the list to contain the *blink* attribute.
- Click on the **blink** attribute in the right-hand list.



Set Driven Key
Load Options Key Select Help
Driver
eyeControl
translateX
translateY
translateZ
worry
blink
Driven
eyelid
makeNurbSphere2
radius
startSweep
endSweep
degree
sections
spans
heightRatio

Set Driven Key window

3 Key the scale positions

- In the Set Driven Key window, click on the **Key** button.

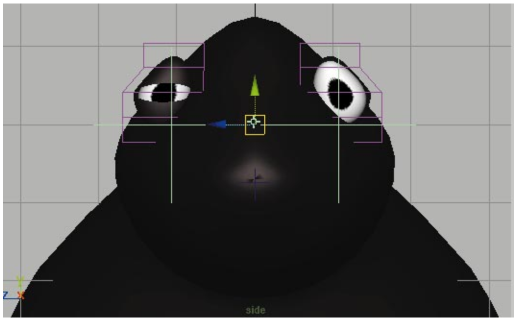
  Now when the **blink** attribute is set to 10, the two sweep angles will be set to their open position.
- In the Channel box, change the **Blink** attribute to **0**.
- Select the *eyelid* node within the hierarchy, then click on the *makeNurbSphere* input node.
- In the Channel box, change the following:

  startSweep to **0**;

  endSweep to **360**.
- In the Set Driven Key window, click on the **Key** button.

4 Test the results

- Select the *eyeControl* group using its selection handle.
- In the Channel box, click on the *Blink* channel name.
- In the Perspective panel, click-drag with the middle mouse button to test the attribute.



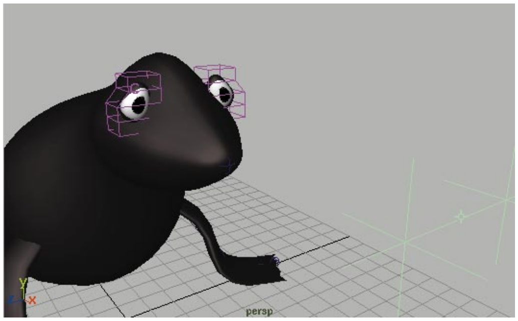
Salty blinking

5 Use Set Driven Key on the left eye

- Set the **Blink** attribute back to **10**.

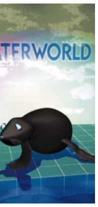
- Select the *eyeLid1* node belonging to the *leftEye* group.
- Load it as **Driven** in the Set Driven Key window then key the relationship between the *blink* and the *eyeLid* node's *startSweep* and *endSweep* angles. This eye should work just like the first eye.
- In the Channel box, test the *Blink* attribute to make sure that both eyes are working properly.
- In the Channel box, set the following:

  Blink to **8**.



Salty with eyes starting to blink

Adding a pupil attribute



You have now linked attributes to the scaling of a cluster and the sweep angles of a sphere input node. Another node type that can be controlled is a texture node.

You will now control the size of Salty's pupils to give you more control over the facial look. You could use Set Driven Key but for comparison, you will use an expression.

1 Add a new attribute

- Make sure that the *eyeControl* group is selected.
- Select **Modify → Add Attribute...**
- In the Add Attribute window, set the following:

  Attribute Name to *pupil*;

  Data Type to **Float**;

  Minimum Value to **0**;

  Maximum Value to **10**;

  Default Value to **5**.
- Click on the **Ok** button.

2 Get the ramp position attribute

- Select **Window → General Editors → Channel Control...**
- Select **Window → Hypershade...**
- Click on the *ramp1* texture from the visor to add it to the selection list.
- From the Non Keyable field, select *colorEntryList[1].position*.
- Click on the **<< Move** button.
- Click on **Close**.

3 Use an expression to drive the pupil

- Select **Window → Expression Editor**.
- In the **Expression name** field, enter *eyePupil*.
- Click on *colorEntryList[1].position* in the **Attributes** list.
- Click in the **Expression:** field and enter the following expression:

```
float $positionStart = 0.9;
float $positionChange = 0.2;
```

```
colorEntryList[1].position =
            $positionStart -
            ($positionChange *
            eyeControl.pupil/10)
```

- Click on the **Create** button.

  This expression is set up as a script. You first set up two variables for the start position of the ramp and the amount you want to change the ramp. What follows is an equation that uses the *eyeControl*'s pupil attribute to drive the change value.

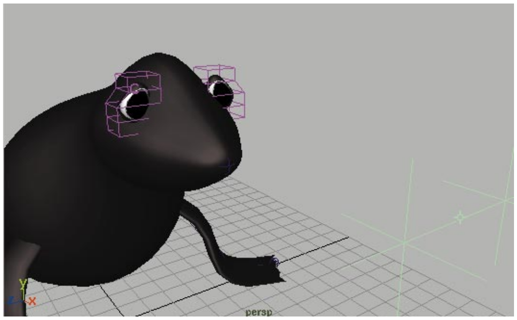
  As you can see, the Set Driven Key lets you generate the same kind of control without figuring out equations or scripts.

Note: This expression uses Maya's MEL (Maya Embedded Language) to set up the expression. You will learn more about MEL and its syntax in Lesson 23.

4 Test the results

- Select the *eyeControl* group using its selection handle.
- In the Channel box, click on the *pupil* channel name.
- In the Perspective panel, click-drag left and right with the middle mouse button.

  The pupil does not update until you release the mouse button.



Salty's pupils at their largest size

5 Save your work

Building the whiskers

For Salty's whiskers, a primitive plane will be used. You will then create and assign a shader that uses a transparency map to create the effect of the whiskers.

Using CV editing techniques which you learned in the last lesson, you will shape the whisker surface.

1 Place a half cylinder

- Go to the **Modeling** menu set.
- Select **Create** → **NURBS Primitives** → **Cylinder - ❐**.
- In the Option window, set the following:

  Axis to **X**;

  End Sweep to **-180**;

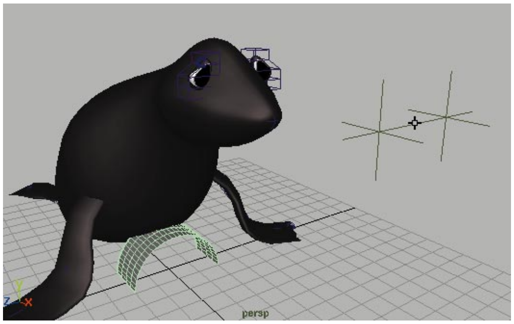
  Radius to **2**;

  Ratio... to **1**;

  Sections to **8**;

  Spans to **1**.
- Click on **Create** and then **Close**.
- Press the **3** key to increase the display smoothness of the shape.

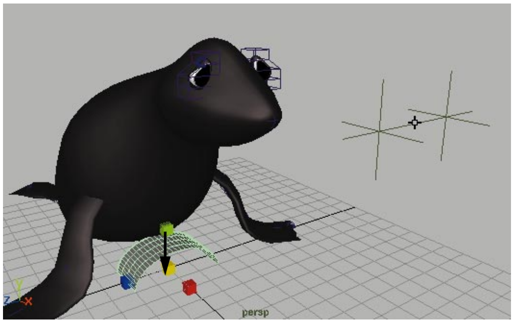
- In the Channel box, rename the *nurbsCylinder* node to *whisker.*



Half cylinder

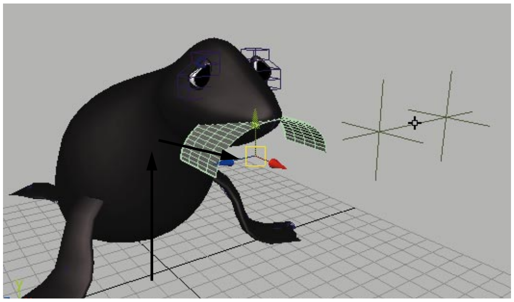
2 Move the whiskers into place

- **Scale** the surface down a little.



Scaled whisker surface

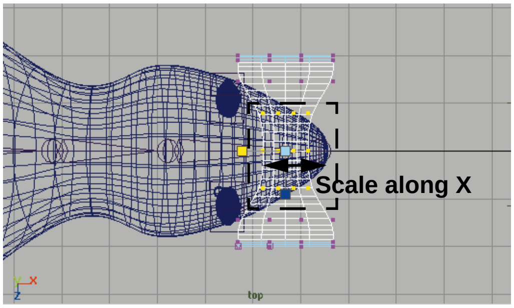
- **Move** the surface until it sits in the proper position on Salty's face.



Moved whisker surface

3 Edit CVs on the shape

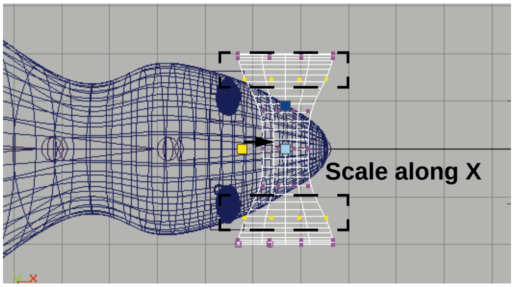
- Press **F8** to go into component selection mode.
- In the Top view, click-drag a selection box around the middle three rows of CVs and scale them along X.



Scaled CVs

- Repeat this for the next two rows of CVs to scale them out a little.

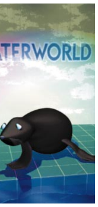
  This creates a more gentle curve at the edge of the whiskers.



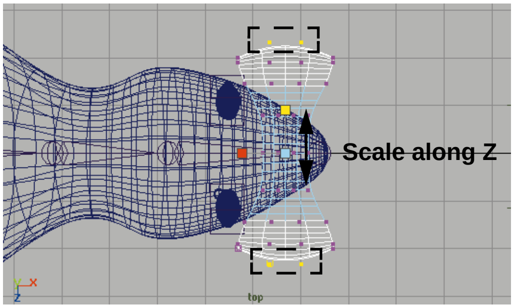
Scaled CVs

4 Scale out the middle CVs

- Click-drag a selection box around the middle two CVs at both ends of the surface.
- **Scale** these out from the center.



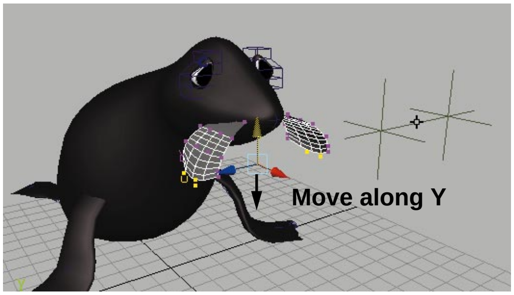
This will add some curvature to the end of the whiskers.



Scaled CVs

- **Move** these down along the Y-axis.

  Now the center of the whiskers droops down a little further than the edges.



Scaled and positioned CVs

- Press **F8** to return to object selection.

Texturing the whiskers

The whiskers have been created with one surface and a texture will be used as a transparency map on the material node to create the lines of the whiskers.

1 Create whisker shading group

Because a transparency map will be used, it is important that the highlights do not appear on the surface. A lambert material node will be used because it does not produce any specular shading qualities.

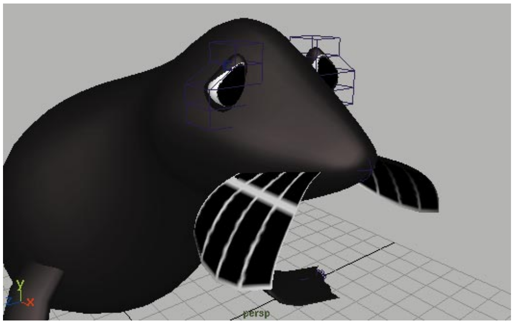
- Select **Window** → **Hypershade...**
- Create a new **Lambert** material node.
- Rename the nodes *whiskerSG* and *lambertWhisker*.

2 Map grid texture onto material color

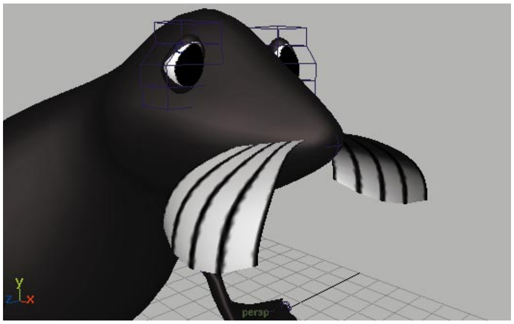
Although the whiskers will eventually be created with just a transparency map, a grid texture will first be mapped onto the color attribute. This means that the adjustments to the texture and placement can be seen in the context of the model view by those who can use hardware texturing.

- Open the *lambertWhisker* material node’s Attribute Editor.
- Click on the **Map** button for the **Color** attribute.
- Click on the **Textures** tab of the Create Render Node window.
- Make sure **With New Texture Placement** is checked on.
- In the **2D Textures** section, click on the **Grid** button.
- **Assign** the *whiskerSG* node to the *whisker* surface.

  If you have texture RAM, you should be able to see the textured surface in the modeling views. If not, you can see the changes to the texture within the Attribute Editor samples and the Hypershade swatches.



Texture on whisker surface



Whisker texture qualities

3 Edit the grid texture's attributes

The grid texture is adjusted to produce black lines on a white surface. The lines in the direction of the whiskers are thickened to become more visible and the cross lines are set to be invisible, transforming the grid into a series of parallel lines.

- Open the *grid1* texture node in the Attribute Editor and set the following:

  Name to *whiskerTexture;*

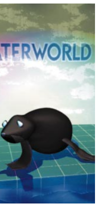
  Line Color to black;

  Filler Color to white;

  U Width to **0.2**;

  V Width to **0**.

  If the lines of the grid texture do not flow as shown, switch the **U Width** and **V Width** settings.



Note: If you do not have texture RAM, you will have to wait until you complete a test rendering later.

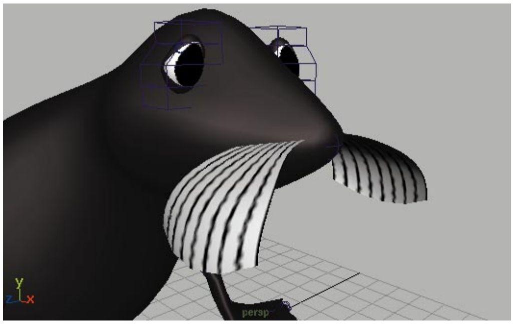
4 Edit the grid texture placement

The number of lines in the grid texture is not set in the texture node itself. Instead the number of lines, or whiskers, is set in the *place2dTexture* node.

- Click on the *place2dTexture* node tab in the Attribute Editor.
- Change the following:

  Repeat U to **8**.

  Because the thickness of the grid lines for V has been set to be invisible, the grid lines along V will be invisible.

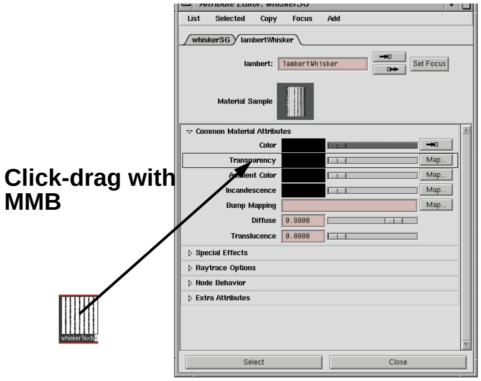


Edited placement attributes

5 Remap grid texture onto transparency

Now that the texture for the whiskers has been developed as a color map, the same texture can be mapped onto the transparency attribute of the same material node. The white areas of the texture will be transparent and the black areas opaque.

- Open the *lambertWhisker* material node's Attribute Editor.
- With the middle mouse button, click-drag the *whiskerTexture* node from the Hypershade onto the **Transparency** attribute in the Attribute Editor.


Click-drag with MMB

lambert: lambertWhisker

Material Sample

Color

Transparency

Ambient Color

Incandescence

Bump Mapping

Diffuse 0.8000

Translucence 0.0000

Assigning texture to tranparency

The one texture node is now mapped onto both the color and transparency attributes of the material node.

Note: Transparency maps cannot be seen by default with hardware rendering. Since the color map and the transparency map share the same placement node, you can use the color map's position to determine the position of the transparency map.

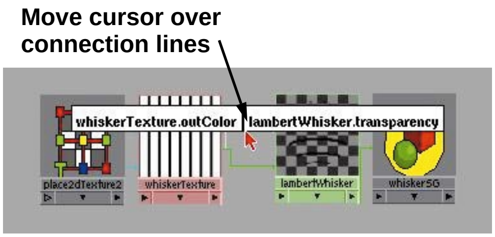
6 View the shading group dependencies

The shading group now contains several nodes — the shading group node, the material node, the texture node and the texture placement node.

The structure of the shading group nodes appears in the open Hypergraph panel. There are two connections between the texture node and the material node which is, in turn, connected to the shading group node itself.

- Move the cursor over each of the connection lines.

  The connected attributes of both the up and downstream nodes are indicated.


Move cursor over connection lines

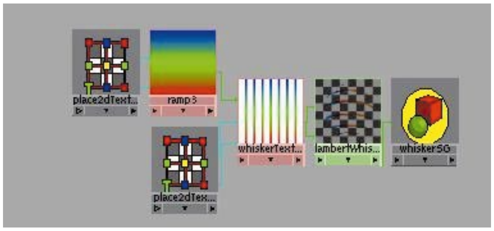
Hypershade view of connections

7 Add a ramp texture to grid texture

In order to add some more detail to the whiskers, another texture map can be added to the shading group. In this case, an embedded texture node can be used. This is when one texture node is connected to an attribute of another texture node. This overlays the effects of the two textures.

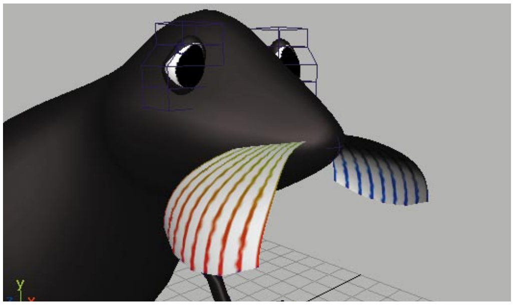
- Open the *whiskerTexture* node's Attribute Editor.
- Click on **Map** next to the **Line Color** attribute.
- In the Create Render Node window, go to the 2D Texture section and click on the **Ramp** button.

  A new texture node and 2D placement node is added to the shading group and connected to the existing texture node, not to the material node.



Shading group with new ramp

The lines of the whiskers now take on the color of the ramp texture.



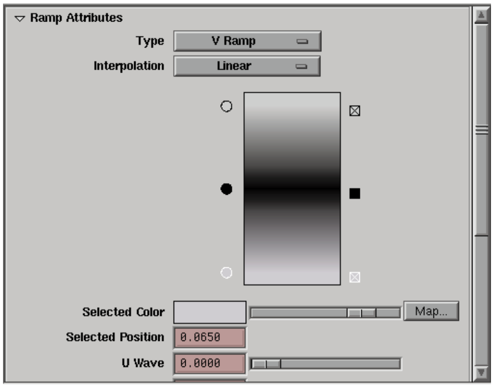
Texture with ramp

8 Edit the ramp texture parameters

The ramp texture can be adjusted so that the whiskers are light at the ends, and dark and heavy closer to the face. Because the whisker surface is one surface that goes through the body of the seal, the ramp should be symmetrical about the center.

- Open the ramp texture node's Attribute Editor.
- Click on the round selector at the center of the ramp.

  The **Selected Position** should be **0.5**.
- Change the **Selected Color** to black.
- The other two selectors should be light gray and positioned as shown.



Ramp Attributes

Type: V Ramp

Interpolation: Linear

Selected Color

Selected Position: 0.0650

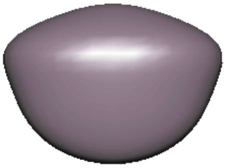
U Wave: 0.0000

Ramp settings

9 Save your work

Building Salty's nose

You will now build Salty's nose using the techniques learned so far. You will CV pull the surface into shape shown below:



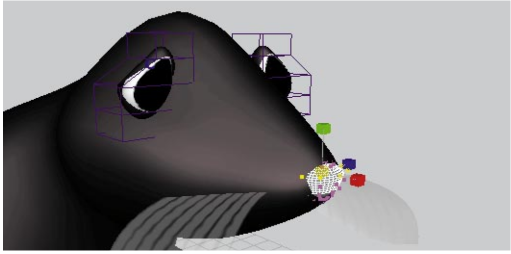
Front view of Salty's nose

1 Place a primitive sphere

- Create a primitive sphere and rename it *nose*.
- **Move** the sphere up to the tip of Salty's face.

2 Manipulate CVs on the shape

- Go to component selection mode.
- Use techniques taught in the last lesson to reshape the sphere into the desired shape as shown below:



Edited CVs

3 Create the nose shading group

The shading group for the seal's nose will be similar to that of the seal's body, but with sharper or harder specular shading qualities to produce the effect of a nose that is hard, wet material and slightly reflective.

- **Create** a new **Blinn** material node.
- Rename the nodes as follows:

  Shading Group node to *noseSG;*

  Material node to *noseBlinn*.
- Open the *noseBlinn* node in the Attribute Editor.
- Change the **Color** to a pale purple.
- In the **Specular Shading** section set the following:

  Eccentricity to below **0.15**;

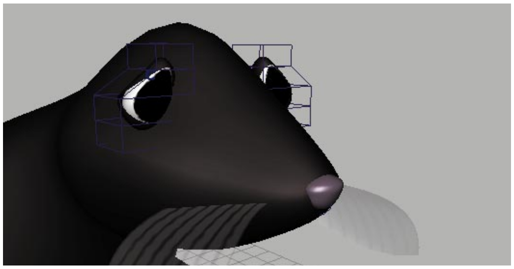
  Specular Roll Off to about **0.8**;

  Specular Color to almost white;

  Reflectivity to **0**.

  The specular highlight on the surface becomes small and sharp, with high contrast against the base diffuse color. Reflectivity has been turned off since this piece is too small to create any useful reflections in the final animation.

- **Assign** the *noseSG* node to the *nose* surface.



Material qualities of the Salty's nose

4 Save your work

Parenting to the skeleton

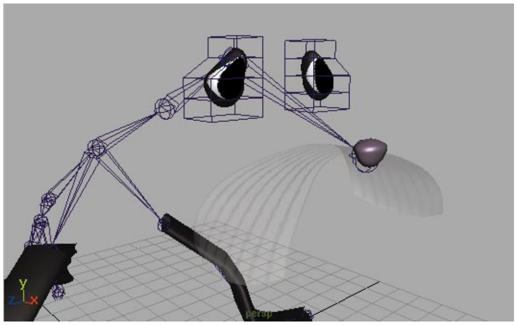
It seems like Salty is built and ready to animate. But since you will want these pieces to move when Salty's head moves, you will need to parent them into the skeleton.

Parenting can be used to add nodes into existing hierarchies. To help you visualize the results, you will use the Hypergraph panel's freeform layout where you can move nodes wherever you need them.

1 Hide the main surface

- Select Salty's main body surface.
- Select **Display → Hide → Hide Selection.**

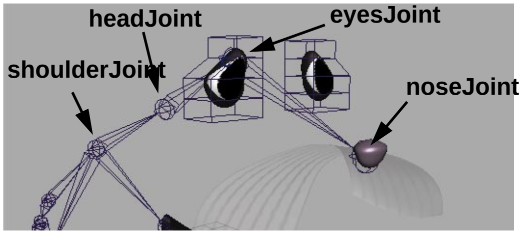
  Now you can more easily see the joints and the other surfaces



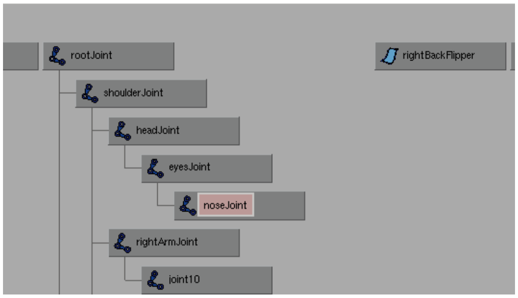
Hidden body surface

2 Rename joints in the Hypergraph view

- In the Hypergraph view, click on the **Scene Hierarchy** button.
- Dolly and track until you see the joints that belong to the head.
- Press the **Ctrl** key and double click on joints to rename them. Rename the head joints as shown below:



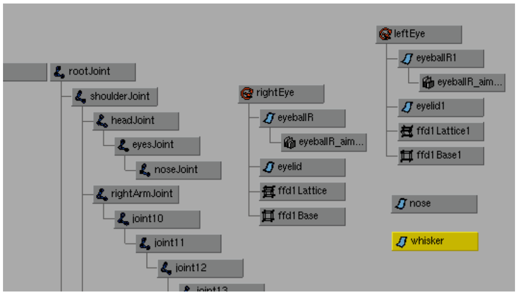
Salty's head joint names



Renamed joints

3 Move the nodes using Freeform mode

- Make the Hypergraph panel full screen.
- Click on the **Toggle Freeform/ Automatic Layout Mode** button.
- Dolly out until you can see all off the nodes.
- With the left mouse button, drag the left and right eyes groups, the *nose* node, and the *whisker* node next to the head joints.

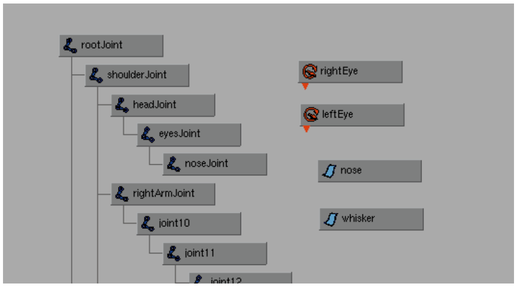


Freeform node layout

- Double-click on the top nodes of the left and right eye groups.
- This collapses the hierarchies into a single node.

This makes it easier to work with these two nodes.

- Click-drag on the nodes with the left mouse button to move them close to the head nodes.

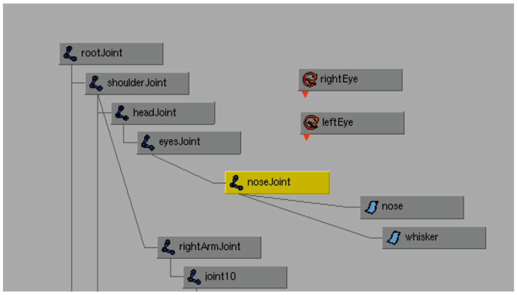


Freeform node layout

4 Parent the whisker to the head

- Click-drag with the middle mouse button on the *whisker* node and drag it onto the *noseJoint* node.
- Click-drag with the middle mouse button on the *nose* node and drag it onto the *noseJoint* node.
- This parents this node to the *noseJoint*.

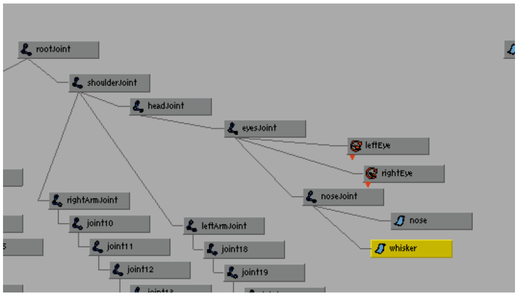
Now the whisker will move when the skeleton is animated.



Parented nodes

5 Parent the eye groups to the head

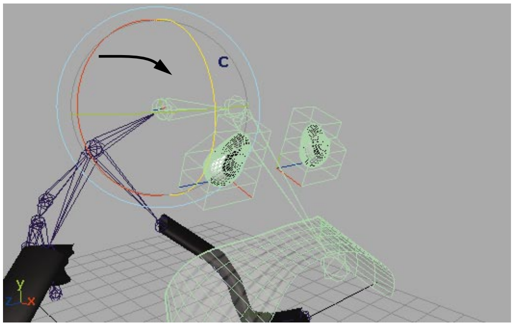
- Click-drag with the middle mouse button on the *rightEye* node and drag it onto the *eyesJoint* node.
- Click-drag with the middle mouse button on the *leftEye* node and drag it onto the *eyesJoint* node.
- This parents these nodes to the *eyeJoint*.



Parented nodes

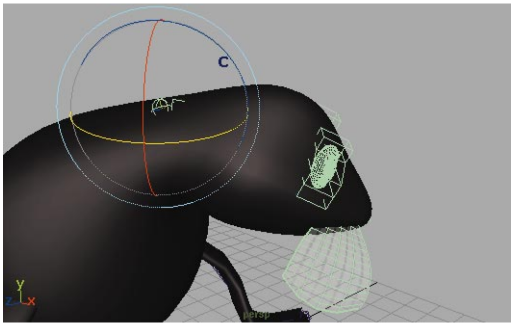
6 Test the results

- Select the *headJoint.*
- **Rotate** it to test the new hierarchy.



Rotated head joint

- Select **Display** → **Show** → **Show Last Hidden**.



Rotated neck joint

- Test the rotation with all of the surfaces.

7 Return to the bind pose

- Select the skeleton.
- Go to the **Animation** menu set.
- Select **Skin** → **Go to Bind Pose.**

  This sets the skeleton is back to its bind pose.

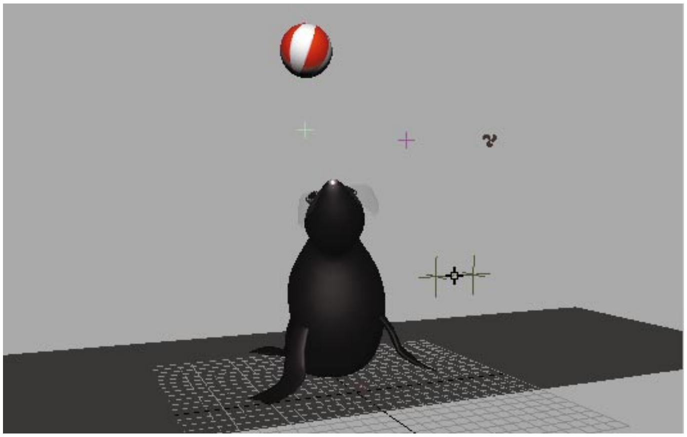
8 Save your work

Conclusion

Salty is now built and ready to animate. In the next lesson, you will add IK chains to Salty's bones, and then you can animate the bouncing ball. You will also begin to take advantage of the extra work used in this lesson to build the eye controls, since keying these features using the new attributes is a straightforward process.

20 Animating Salty

In this lesson, you will make Salty bounce a ball on her nose. At first you will set up *Inverse Kinematic* (IK) chains that offer you control over the motion of Salty's bones. You will also animate the movement of the eyes so that they follow the ball around the scene.



Salty on the move

In this lesson you will learn the following:

- How to set up IK chains
- How to animate the IK chains
- How to constrain a bouncing ball to Salty's nose
- How to animate the eyeControl group
- Soft body dynamics to animate whiskers

Initial set-up

To start this lesson, you will continue with the file from the last lesson.

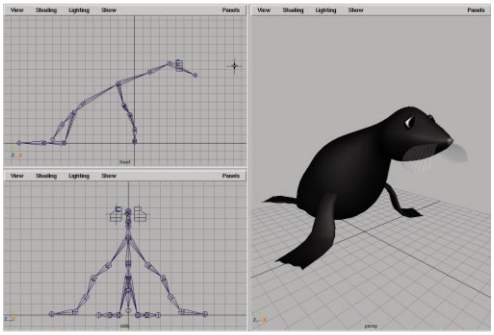
1 Change the panel layout

- Select **Panels → Layouts → 3 Left Split**.
- Set the top left window to a Front view, the bottom left window to a Side view, and the right view to a shaded Perspective view.
- Make the Perspective view the active panel and press the **5** key to set smooth shading.

2 Use the Show menu to set up views

- In the Front view panel's **Show** menu, turn **Surfaces** to **Off**.
- Repeat this step for the Side view.

  This lets you focus on the skeleton in these views while still seeing the surfaces in the Perspective view.


View Shading Lighting Show Panels
View Shading Lighting Show Panels
View Shading Lighting Show Panels

Recommended panel layout

Adding IK single chain handles

To help control Salty's flippers, you need to build *inverse kinematic* (IK) chains. These will let you control the flippers while Salty bounces the ball with her nose.

To animate a joint hierarchy, you could rotate each joint individually to set poses for the character. But, this can be time-consuming if you were to set keys for all of the joints.

Instead, you will use an *IK chain*. With an IK chain, you define a start joint and an end joint. The start joint is known as the *root* of the chain while the end joint has a corresponding *end effector*. You define the end effector by placing an *IK handle* that is then used to control all the bones within the chain.

1 Set the IK handle options

Maya offers you several types of IK chains. For the flippers, you will use an IK single chain solver which is the simplest IK solver.

- Go to the **Animation** menu set.
- Select **Skeleton → IK Handle Tool - ❐**.
- In the option box, set the following:

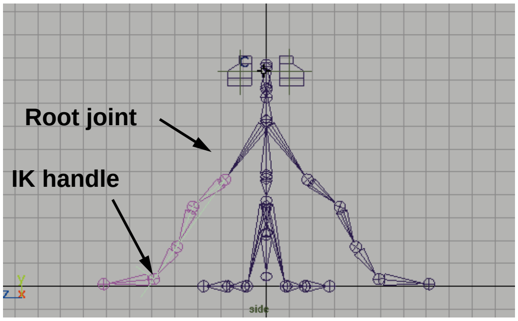
  Current Solver to **ikSCsolver.**
- This tool is now active. Click **Close**.

2 Add IK handles to the right flipper

IK chains are defined by a root joint where the chain starts, and an IK handle where the chain ends. This handle can then be used to control the chain.

- Click on Salty's right shoulder joint to set the root of the IK chain.
- Click on the second-last joint on Salty's flipper to place the IK handle.

The chain is drawn between these two joints.

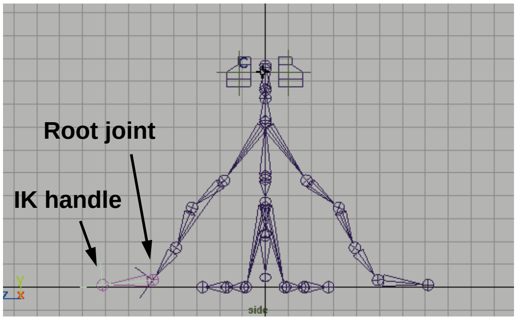


First IK chain

3 Add a second IK handle to the flipper

- Press the **y** key to reinvoke the **IK Handle Tool**.
- Click again on the second-last joint to start another IK chain.
- Click on the last joint on Salty's flipper to place the IK handle.

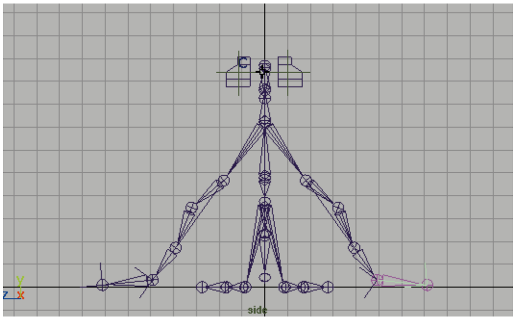
  This IK chain will be used to help keep Salty's flipper on the ground as Salty strains to bounce the ball.



Second IK chain

4 Add similar IK chains to the left flipper

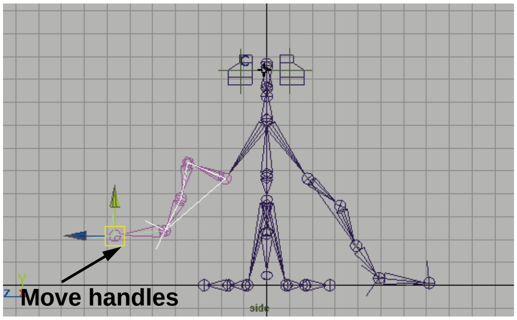
- Use the same technique to add two similar IK chains to the left flipper.



Left flipper IK chains

5 Test the IK chains on the right arm

- Select the two IK handles on the right flipper, making sure that you mask everything except the handles.
- **Move** the flipper in the Side view and watch the results in the Perspective view.
- All the joints in the flipper hierarchy are moved as one. As you move the IK handles, the interior joints bend.



Translated IK handles

- Press the **z** key to **Undo** this move.

Rather than moving the legs during this lesson, you want them to stay on the ground. Later, you will lock them down so that they don't accidentally move on you.

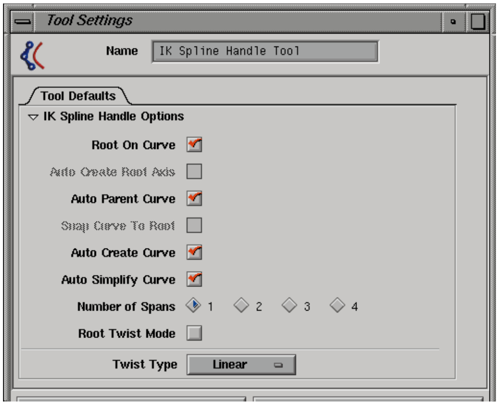
Add an IK spline handle

The *IK Spline* solver is another type of IK solver that lets you use a curve to control the joint rotations. You can then animate the skeleton by either moving the joints along the curve, or by animating the position or shape of the curve.

This kind of IK chain is a good choice for Salty's neck where most of the animation in this lesson will take place.

1 Set the spline IK options

- Select **Skeleton** → **IK Spline Handle Tool - ❐**.
- In the option box, make sure that the following options are set:



Tool Settings

Name: IK Spline Handle Tool

Tool Defaults

IK Spline Handle Options

Root On Curve ☑

Auto Create Root Axis □

Auto Parent Curve ☑

Snap Curve To Root □

Auto Create Curve ☑

Auto Simplify Curve ☑

Number of Spans ● 1 ○ 2 ○ 3 ○ 4

Root Twist Mode □

Twist Type: Linear

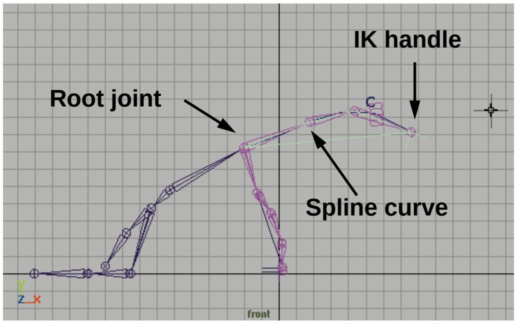
IK handle tool options

- Click **Close**. This tool is still active.

2 Add an IK spline handle

- Click on the joint at the nape of Salty's neck as the root joint.
- Click on Salty's nose joint as the end joint.

  An IK chain is drawn between these joints. At the same time, a curve has been placed along the joints. You can use the panel's **Show** menu to see this curve.

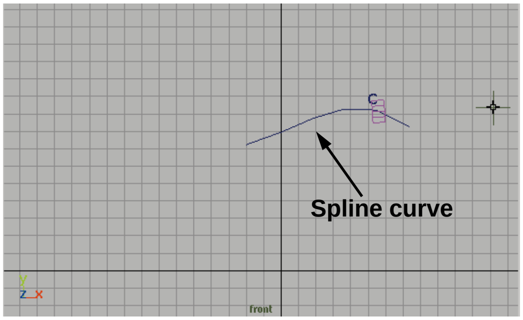


IK spline chain

3 Show the curve in the Front view

- Change the Side view to another Front view panel.
- In the new Front view panel's **Show** menu, turn **Joints** and **IK Handles** to **Off**.

  Now you can use this window to select and transform the curve itself, while viewing the effect on the joints in the other views.

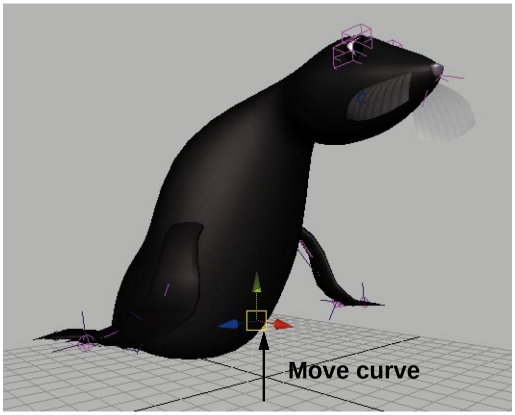


IK spline curve

4 Test the effect of the new chain

- **Select** the new curve.
- **Move** the new curve up in the Front view.

  The whole head and neck moves as one, bringing the flippers along.



Moving the spline curve

- **Undo** this move.

5 Set a key on the flippers

- In the Perspective window, **Select** all four of the IK handles belonging to the flippers.
- **Move** the IK handles along the Y-axis to raise the flippers above the ground level just a little.
- Set the **Time slider** to frame **1**.
- Press **Shift w** to set keys on only the translation channels belonging to the IK handles.

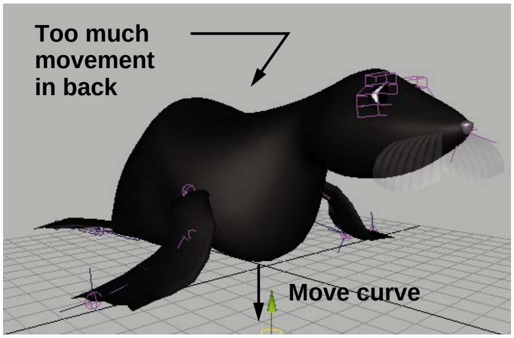
  Setting keys helps to lock down the flippers, which helps when you move Salty's neck later.

Tip: To set keys for only translation, rotation or scaling remember the following hotkey combinations:
Shift w - Translation
Shift e - Rotation
Shift r - Scaling

6 Re-test the chain

- **Select** the new curve.
- **Move** the new curve down in the Front view.

  As the head and neck moves down, the flippers stay planted on the ground.



Curve moved with flippers keyed

- **Move** the curve up.

The flippers try to stay planted until Salty goes beyond their reach. Now, you can focus on the fact that Salty's back is deforming in an uncomfortable manner.

- **Undo** these two moves.

Cluster the spline curve

In order to get more control over Salty's motion, you will place a *cluster deformer* onto the curve, then adjust the cluster weights of the CVs to refine how they will animate.

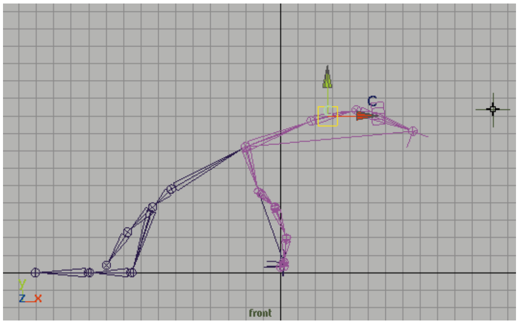
You could now animate the joints by setting keys for the CVs of the new spline curve, but a cluster will offer better control because of the ability to weight the cluster points.

1 Cluster the curve

By adding a cluster to the new curve, you will be placing the curve's CVs into a special cluster set.

- Select the spline curve.
- Select **Deform → Create Cluster**.

  A cluster is now created using the spline's CVs. The cluster is indicated using a small 'c' icon. This represents a cluster handle node.



New cluster

2 Edit the cluster weights

One key feature of a cluster is the ability to set weights for each CV in the cluster. For Salty, you will use a lighter weight for the CVs near Salty's neck and a heavier weight at her nose. This means that her head will move more than her neck, thereby giving you more subtle control as to how the head moves.

- Select the spline curve.
- Press **F8** to enter Component mode.
- Select the four CVs for the curve.
- Select **Windows → General Editors → Component Editor...** and click on the *Weighted Deformers* tab**.**
- Press **Load Components** in the Editor.

  The four CVs are now listed with their current weight setting of 1.0.

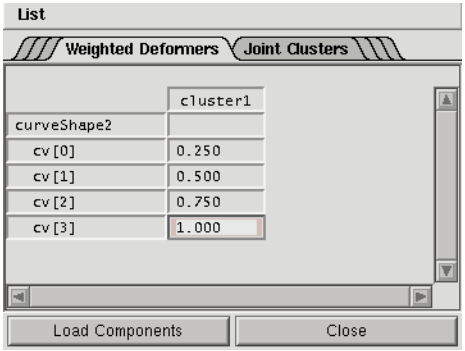
- Enter the following values for the CVs:

  cv[0] to **0.25;**

  cv[1] to **0.5;**

  cv[2] to **0.75;**

  cv[3] to **1.0;**



	cluster1
curveShape2	
cv[0]	0.250
cv[1]	0.500
cv[2]	0.750
cv[3]	1.000

Weighted CVs

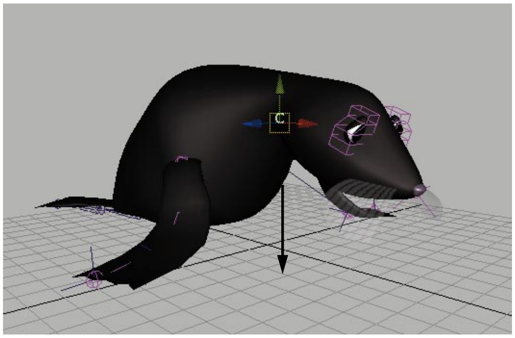
- Close the Component Editor.
- Press **F8** to go back to Object mode.

3 Test the results

Now that the CVs cluster weights have been set, you can move the cluster to see the advantage of using the weights.

- Since one of the CVs is now the current selection, once again **Select** the cluster handle using the small 'c' icon.
- **Move** the cluster down along the Y-axis.

  The head and the flippers move.

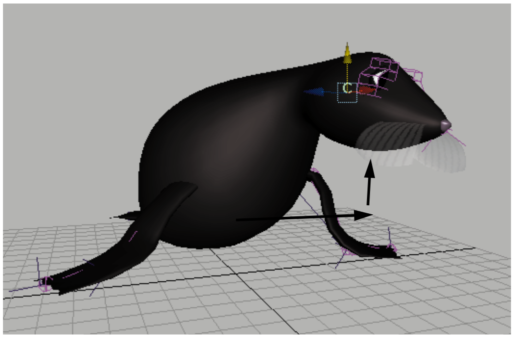


Moved cluster

- **Move** the cluster out along the Z-axis using the manipulator in the Perspective view.

  The cluster is helping the head move in a more graceful manner. The cluster makes sure that the neck remains fairly stationary while the head moves more freely.

  As the head moves, you may notice that a little bit of motion is placed on the neck because the first CV in the IK spline curve has been given a weight of 0.25. Generally, it is a good idea to *not move* the first CV of an IK spline curve. For Salty, a little bit of motion is used because of Salty's more limited range of motion.



Moved cluster

- **Undo** these moves until you return to Salty's start position.

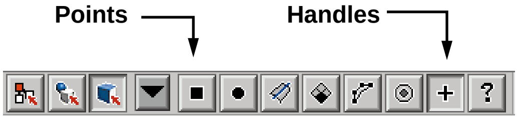
4 Display the cluster's selection handle

- With the cluster handle selected, select **Display → Object Components → Selection Handles**.

This toggles on the cluster node's selection handle. Every node in Maya has a selection handle that can be revealed to simplify selection later.

5 Move the selection handle

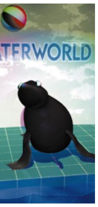
- Press **F8** to go into component selection mode.
- Turn off the points selection mask and turn on the handles mask.

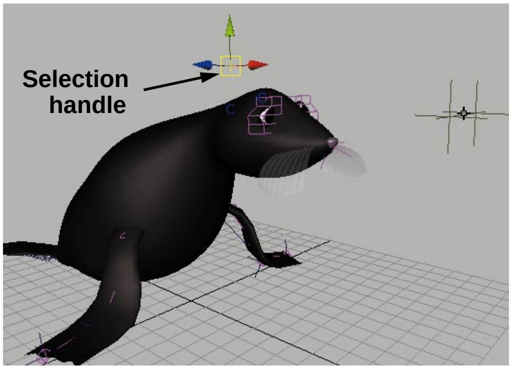


Selection masks

- Click-drag a selection box around the cluster's selection handle to pick this component.
- **Move** the handle along the Y-axis to place it above Salty's head.

  It will now be easier to find later.





Selection handle

- Turn off the handles selection mask and turn on the points mask.
- Press **F8** to go back to object selection mode.

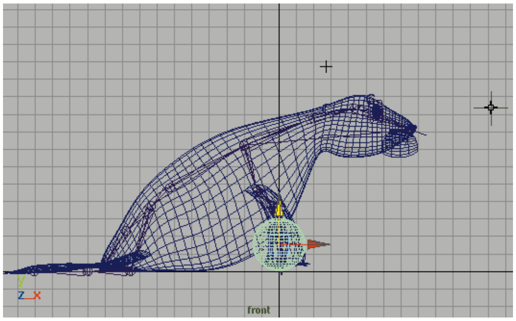
Create a ball

To animate Salty bouncing a ball, you need to create the ball itself. This ball will use a simply hierarchy of grouped nodes. One node will be used to animate the bounce of the ball while the top node will be used to constrain the ball to Salty's nose.

1 Create a sphere

- Select **Create** → **NURBS Primitives** → **Sphere**.
- Name this node *ball*.
- Press **3** to increase the surface smoothness.
- Click on the input node in the Channel box and set the following:

  Radius to **1.5**.
- **Move** the sphere up until it sits just above the ground plane.



Primitive sphere

2 Delete history

- Select **Edit** → **Delete by Type** → **History**.

3 Group the sphere to itself

This will create a node hierarchy for your animation. One node will be used to animate the ball's bounce while the top node will be constrained to Salty's nose.

- Select **Edit → Group**.
- Name this node *ballConstrain*.

4 Create a ball shading group

- In the Hypershade, create a Phong material node.
- In the Channel box, rename this node *ballPhong*.

5 Add a texture to the shading group

- Press **Ctrl-a** to open the Attribute Editor.
- Click on the **Map** button next to **Color**.
- Choose a checker texture from the Create Render Node window.
- In the *checker1* node tab of the Attribute editor, change the two colors to red and white.
- Click on the *place2DTexture* node and set the following attribute:

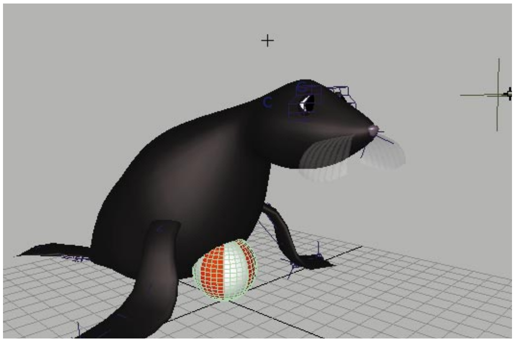
  RepeatU to **0.5**.

6 Rename the shading node

- In the Hypershade, select the *ballPhong* node.
- Click the Show Up and Downstream connections button.
- Press the **Ctrl** key and click on the *phong1SG* node. Rename this node *ballSG*.

7 Assign the shading group

- With your middle mouse button, click-drag the **ballSG** node onto the new sphere.



Ball with checker texture

8 Save your work

Connect the ball to Salty's nose

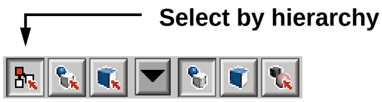
You now want the ball to move with Salty's nose as she gets ready to start bouncing it. A point constraint will be applied to the top node of the ball group.

1 Constrain the ball to Salty's nose joint

- Go to the **Animation** menu set.
- In the Front view panel, select **Show → None,** then **Show → Joints**.
- **Select** the nose joint of the skeleton.

  Only the round joint icon at the end of the skeleton should be highlighted. The first bone should *not* be highlighted.
- Click on the **Select by hierarchy** button.

PROJECT FIVE



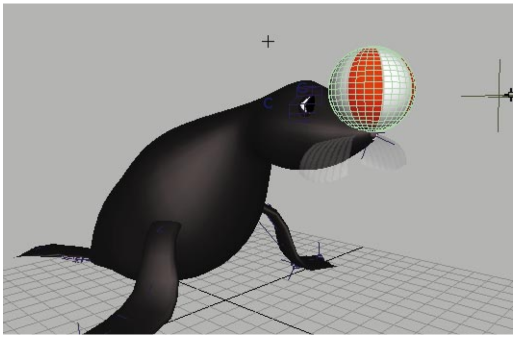
Pick masks

- Press the **Shift** key then **Select** the ball.

  The select by hierarchy mode forces your selection to the top node of the ball hierarchy. The *ballConstrain* node is the node that you will use to constrain to the nose.

- Select **Constrain** → **Point**.

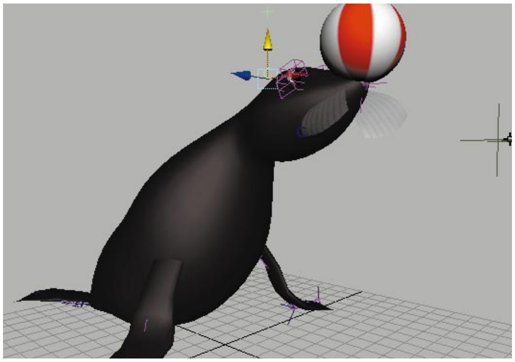
  Now the ball is constrained to Salty's nose.



Ball constrained to Salty's nose

2 Move the cluster to balance the ball

- Press **F8** to go back to object selection mode.
- **Select** the cluster handle using its selection handle.
- **Move** the cluster up until Salty appears to be balancing the ball.



Moving the cluster handle

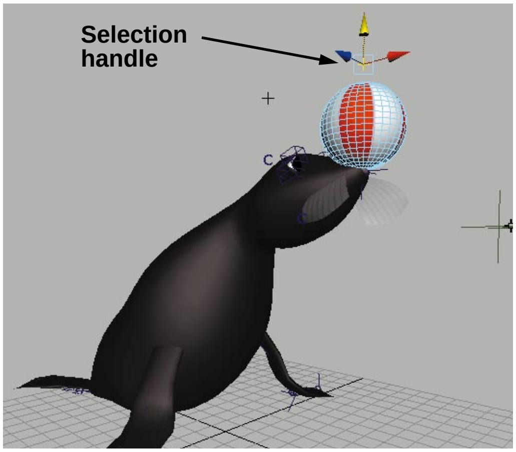
3 Put a selection handle on ball node

You will now place a selection handle on the *ball* node to make it easier to select.

- **Select** the *ball* node.

  Use the Channel box to confirm that the correct node is selected.

- Select **Display** → **Object Components** → **Selection Handles**.
- Press **F8** to go into component selection mode.
- Turn off the points selection mask and turn on the handles selection mask.
- Click-drag a selection box around the ball's selection handle to pick this component.
- **Move** the handle along the Y-axis to place it above Salty's head.



Ball selection handle

- Turn off the handles selection mask and turn on the points mask.
- Press **F8** to go back to object selection mode.

Setting up for the animation

To get ready to animate the scene, you will set up the animation preferences, refine the display of Salty in the Perspective view, and use the channel control to make it easier to set keys on the *clusterHandle* and *ball* nodes.

1 Set the length of the animation

In the animation preferences, you can set the frames per second to 30 and the length of the animation to 450 frames. This will give you a 15-second animation.

- At the right end of the Time slider, click on the animation preferences button.
- Click on the **Units** tab and set the following:

  Time to **NTSC (30 fps)**.

- Now click on the Animation tab and set the following:

  Playback Start/End to **1** and **450**;

  Animation Start/End to **1** and **450**;

  Playback Speed to **Normal**.

  Now the length of your animation is 450 frames long and play back will be at 30 frames per second.

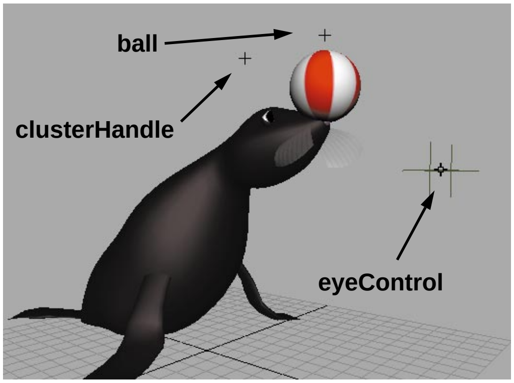
- Click on **Save**.

2 Prepare the Perspective view

It is a good idea to focus on how the surfaces look when animated.

- In the Perspective view panel, select **Show** → **Joints**, **Show** → **Deformers**, and **Show** → **IK handles** to turn these object types off for this window.

  You can now use the three selection handles to quickly select and key the various parts of Salty.



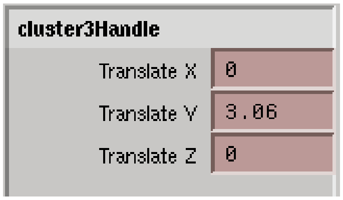
Selection handles in place

Note: By turning off deformers, you have hidden Salty's cluster handle. This is not a problem because you can use the selection handles to pick and animate this node.

3 Restrict the cluster handle channels

To make sure that you are keying the correct channels, you will make some of the attributes non-keyable on the *clusterHandle* and *ball* nodes.

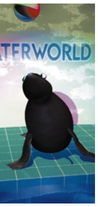
- **Select** the cluster handle for Salty's neck using its selection handle.
- Select **Window** → **General Editor** → **Channel Control**.
- Press the **Ctrl** key and click on all the *rotate* and *scale* channels, and then the *visibility* channel.
- Press the **Move >>** button to make these channels non-keyable.



cluster3Handle

Translate X 0
Translate Y 3.06
Translate Z 0

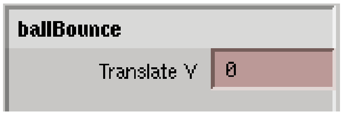
Keyable channels



4 Restrict the ball channels

- **Select** the *ball* node using its selection handle.
- In the Channel Control window, make the *translate X*, *translate Z*, the *rotate* and *scale* channels, and the *visibility* channel all non-keyable.

This leaves only the *translate Y* channel. This is the only channel you will be keying on the *ball* node.



ballBounce

Translate Y 0

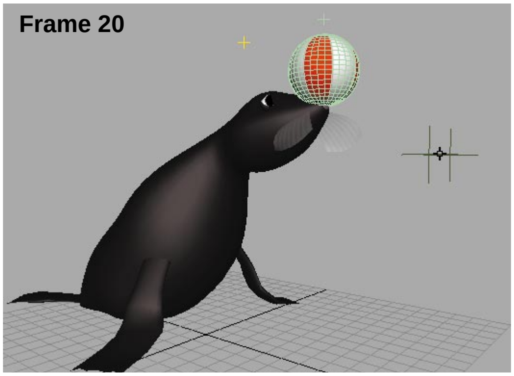
Keyable channels

Animating Salty

It is now time to get Salty moving. You will start by keying Salty's neck using the cluster handle and then you will animate the ball bouncing. You will start with a single bounce, then you will copy and paste these keys to extend the animation.

1 Set initial keys

- Set the **Time slider** to frame **20.**
- **Select** both the *ball* node and the *clusterHandle* node using the two selection handles.
- Press the **s** key to set keys for the keyable channels.



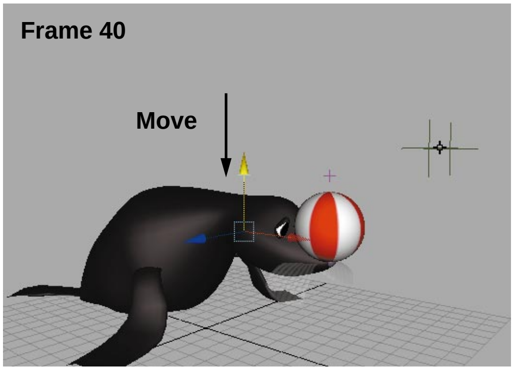
Initial position

2 Set keys on Salty's cluster handle

- Set the **Time slider** to frame **40.**

- **Select** only the *clusterHandle* node using its selection handle.
- **Move** Salty's neck down along the Y-axis.
- Press the **s** key to set keys for the keyable channels.

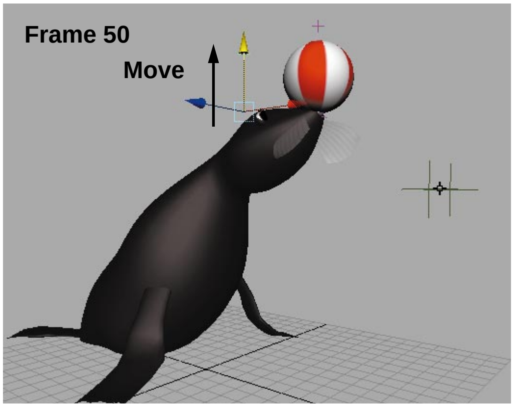
  This drop is where Salty builds the momentum to start bouncing the ball. This action is designed to give the audience a clear indication that Salty is going to bounce the ball. In traditional animation, this is called *anticipation*.



Second key position

- Set the **Time slider** to frame **50.**
- **Move** Salty's *clusterHandle* up along the Y-axis.
- Press the **s** key to set keys for the keyable channels.

  Salty is now pushing the ball up to project it into the air.



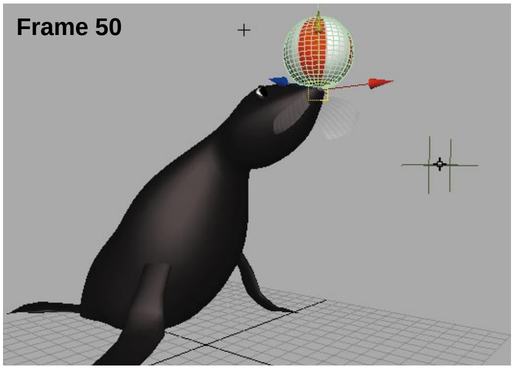
Third key position

3 Set keys on the ball

Salty has been pushing the ball up to project it into the air. You will now key the ball's motion in the air.

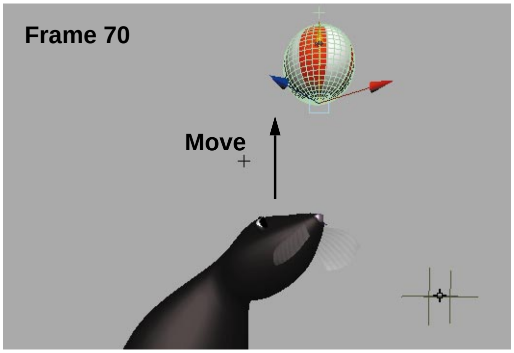
- **Select** only the *ball* node using its selection handle.
- Press the **s** key to set keys for the keyable channels at the current frame.

  This key makes sure that the ball is touching Salty's nose as it is being pushed off.



Second key position

- Set the **Time slider** to frame **70.**
- **Move** the *ball* node up along the Y-axis.
- Press the **s** key to set keys for the keyable channels.



Third key position

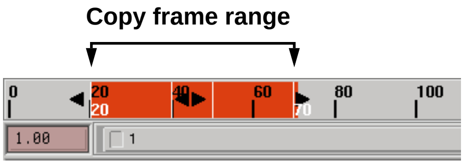
- Click-drag on the Time slider to scrub through the roughed out sequence.

4 Copy a range of keys

To make Salty bounce the ball three times, you will copy and paste the existing keys to extend the animation.

- **Select** both the *ball* node and the *clusterHandle* using the two selection handles.
- Set the **Time slider** to frame **20.**
- Press the **Shift** key and in the **Time slider,** drag to frame **71.**

  A red selection bar appears to indicate the selected time range. The start and end frames of the selection are displayed in white.



Selected keys

- Click in the Time slider with your right mouse button and select **Copy** from the pop-up menu.

5 Paste the keys

- Set the **Time slider** to frame **90.**
- Click in the Time slider with your right mouse button and select **Paste** → **Paste** from the pop-up menu.

New keys have been placed starting from frame 90.

- Set the **Time slider** to frame **160.**
- **Paste** → **Paste** the frames again at this frame.
- Play back the results.

6 Select the animation curve for the neck

- Open the Graph Editor.

Note: You can open the Graph Editor either in one of the existing view panels using its **Panels** menu or as a floating window using the **Window** menu. In earlier lessons, you learned both methods. Now you can begin using the method that you like the best.

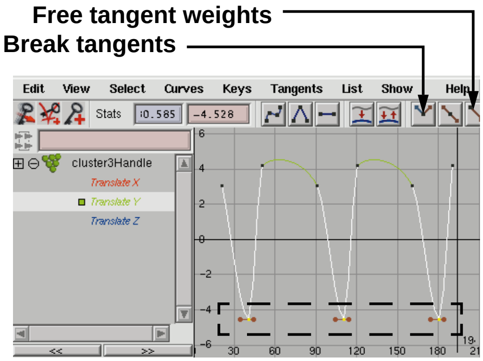
- From the Graph Editor, select **View** → **Frame All.**

- Click on the name *Translate Y* belonging to the *clusterHandle*.

  This isolates this curve from the others.

7 Edit the curve tangents

- Select the bottom three keys on the curve.
- From the Graph Editor, select **Tangents → Flat**.



Free tangent weights

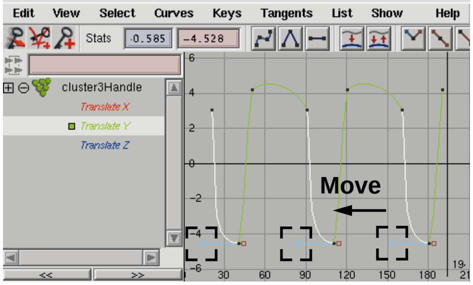
Break tangents

New tangent setting

- Select **Curves → Weighted Tangents**.
- Click on the **Break tangents** button and then on the **Free tangent weight** button.

  The first button lets you break the tangents to create a sharp reaction. The second button lets you refine your curves to make them extra steep.
- Press the **Shift** key and select the tangent handles to the left of each of the selected keys.
- Select the **Move** tool.
- Click-drag with the middle mouse button to drag the tangents out to increase their weight.

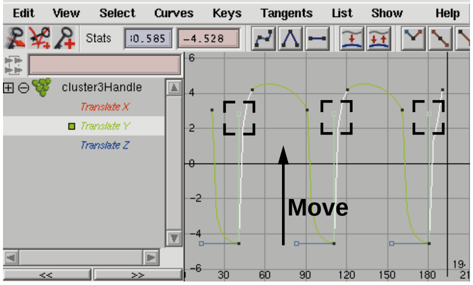
  This curve indicates that Salty will slow down to a stop as she drops her neck down.



Tangent handle edits

- Press the **Shift** key and drag a selection box over the three keys to deselect the tangent handles.
- Press the **Shift** key and select the tangent handles to the right of each of the selected keys.
- Click-drag with the middle mouse button to drag the tangents up to a vertical position.

  This curve indicates that Salty will quickly snap her neck up to project the ball into the air.



Tangent handle edits

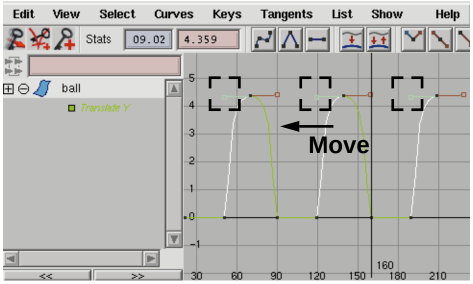
- Play back the results.

  As the curves suggest, Salty slows down as her head drops then speeds up in order to toss the ball.

  Always remember the importance of using the animation curve tangents to refine your motion. The tangent controls help define what happens before and after your keys.

8 Edit the animation curve for the ball

- Click on the name *Translate Y* belonging to the *ball.*
- Select all of the keys at the bottom of the curve.
- Click on the **Linear Tangents** button.
- Select all the keys at the top of the curve.
- From the Graph Editor, select **Curves → Weighted Tangents**.
- From the Graph Editor, select **Tangents → Flat**.
- Click on the **Free tangent weight** button.
- Press the **Shift** key and select the tangent handles on the left side of the keys.
- **Move** the handles to change the weight at this point.



Ball animation curve edits

- Playback the results.

9 Delete the last key on the ball

Since the new locator will be controlling the animation as the ball leaves Salty's nose, the key on *ball* after frame 190 can be removed.

- Select the *ball* node.
- In the Graph Editor or Timeline, delete the key at frame 210.

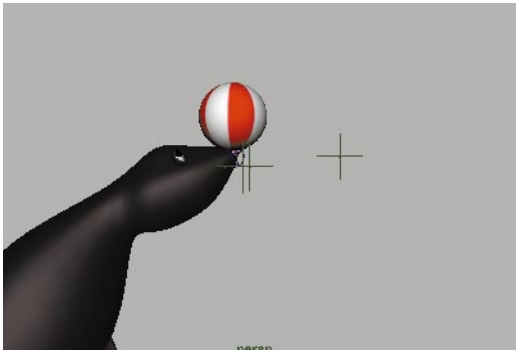
10 Save your work

Bounce the ball off the floor

For Salty's last bounce of the ball off the nose, you will be constraining the ball to another locator. You will then animate the new locator going off screen.

1 Create a new Locator

- Go to frame **190**.
- Select **Create → Locator.**
- Use the Channel Control editor to have only the Translation channels keyable.
- Place the Locator in front of the ball.

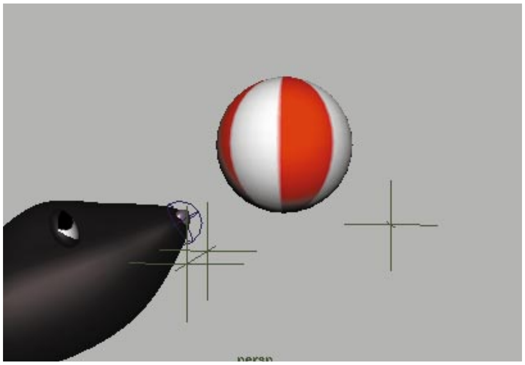


New Locator in front of ball

2 Constrain the ball to the locator

- Select the new Locator.
- **Shift-Select** the *ball* node.
- Press the up arrow to move up the ball's hierarchy to the *ballConstrain* node.
- Select **Constrain** → **Point.**

  The ball is nose floating half way between Salty's nose and the locator. This is because both the constrains have equal weighting.



The ball with two constrains applied

3 Key the Constraint weights

- Select the *ballConstrain* node.
- Select *ballConstrain_pointConstraint* input node and set the following:

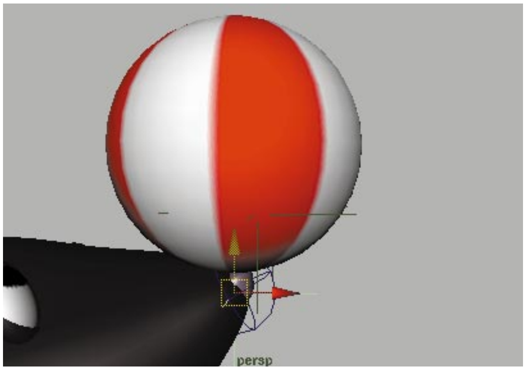
  NoseJoint W0 to **0;**

  Locator3 W3 to **1;**

  The ball is now favoring the nose.
- Go to frame 189
- Select **Key Selected** for the *noseJointW0* and the *Locator3W3* attributes.

4 Move the new Locator

- Select the new Locator.
- While holding the **v** key, point snap the new locator to the end joint at salty's nose.



The locator point snapped to end joint

5 Key the Constraint weights

- Go to frame 190.
- Select the *ballConstrain* node.
- Select*BallConstrain_pointConstrain1* input node and set the following:

  NoseJoint W0 to **0;**

  Locator3 W3 to **1;**

- Click on the Channel box with your right mouse button and select **Key Selected**.

Animate the Locator

You will now animate the locator exiting the scene. Since the weight of the constraint is fully on the locator and not the joint, the ball will follow the locator. You will also be adding a floor under Salty.

1 Create a floor

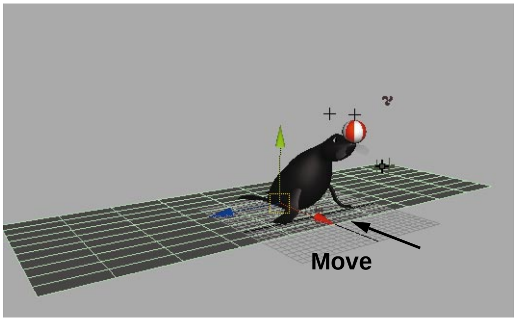
You will create a flat polygon surface for the platform which Salty will perform on.

- Select **Create** → **Polygons Primitives** → **Plane**.
- Rename the surface *deck*.
- Click on the *polyPlane* input node and set the floor as follows:

  Width to **25**;

  Height to **75**.

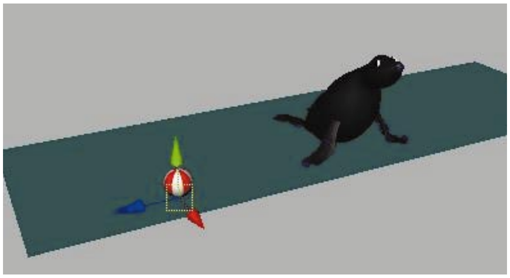
- **Move** the surface along the negative X-axis so that it is positioned with Salty is at its front edge.



Polygon plane moved to position

2 Set a key on the locator

- Go to frame **190**.
- Select the new locator.
- Press **s** to set a key on the translation channels.
- Go to frame **305**.
- Move the locator to the floor and beside salty.
- Set a key for the locator at this position.



The ball on the floor at frame 305

The ball is going straight from the nose to the floor. You will need a key in between where the ball is in the air.

- Go to frame **240**.
- Translate the ball in **Y** over Salty's head.
- Set a key.

3 Refine the motion

- Continue adding keyframes to make the ball bounce more naturally. Use the Graph Editor to modify the curves.

4 Add roll to the ball

To add some rotation to the ball you will add the Rotation Channels back to the *ball* node.

- Select the *ball* node.

- With the Channel Control editor, make the rotation channels keyable.
- Go to frame 305.
- Key rotations on the ball as it bounces off screen.

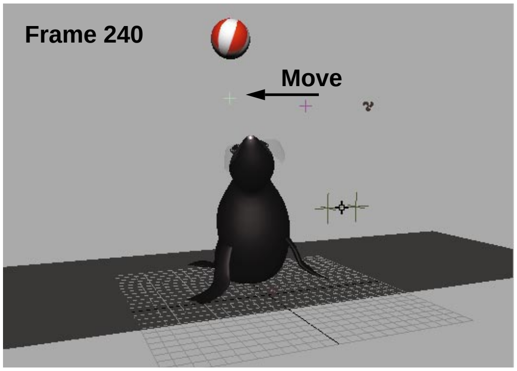
5 Save your work

Finishing Salty's motion

Salty is sure to be upset that the ball is bouncing away out of her control. You will now set more keys to animate Salty's reaction.

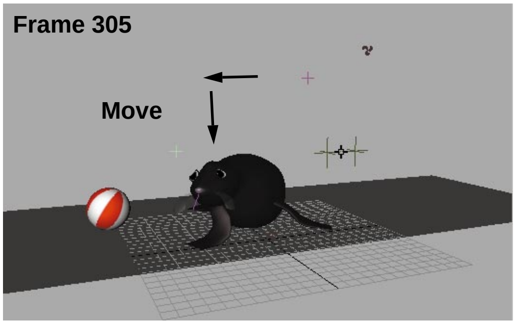
1 Key Salty's cluster handle node

- Select the *clusterHandle* node using its selection handle.
- Go to frame **240**.
- **Move** the *clusterHandle* along the Z-axis to begin following the ball.
- Press the **s** key to set keys for the keyable channels.



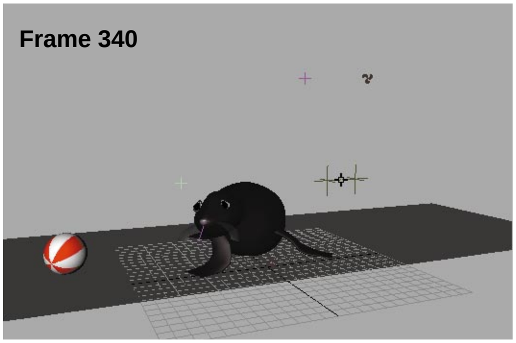
Salty starts to follow ball

- Go to frame **305**.
- **Move** Salty's cluster handle along the Z-axis then down along the Y-axis so that Salty is looking at the ball.
- Press the **s** key to set keys for the keyable channels.



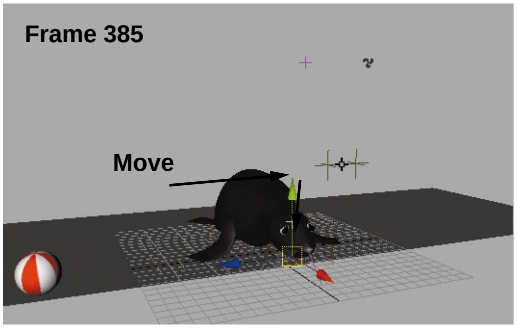
Salty watching ball bounce

- Go to frame **340**.
- Press the **s** key to set keys for the keyable channels.



Salty continues to watch ball bounce away

- Go to frame **385**.
- **Move** Salty's cluster handle back along the Z-axis then down a little further along the Y-axis so that Salty is looking out at the audience in a dejected pose.
- Press the **s** key to set keys for the keyable channels.



Salty in dejected pose

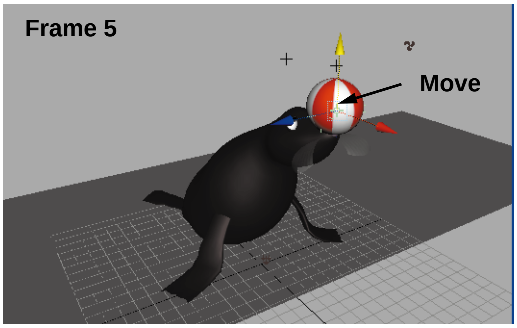
- Play back the results.

Animating Salty's eyes

Earlier, you built a special *eyeControl* node that is designed to control where Salty is looking and other important eye features. Now you will animate this node to make use of this control.

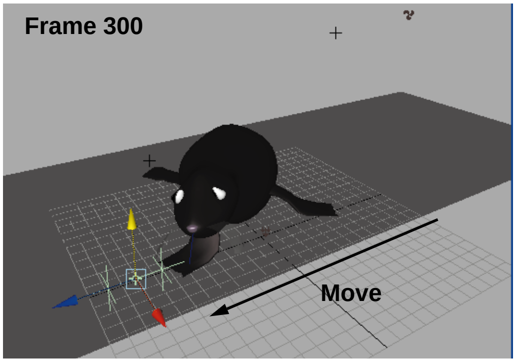
1 Animate the eyeControl position

- Go to frame **5**.
- Select the *eyeControl* group using its selection handle.
- **Move** it to be in the same position as the ball.



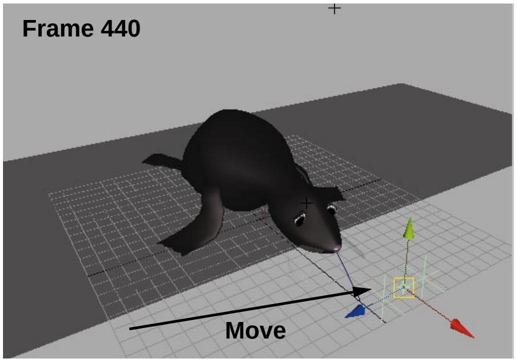
Eye control node positioned

- Press **Shift w** to key the translation channels of this node.
- Go to frame **190**.
- Press **Shift w** to key the translation channels of this node.
- Go to frame **300**.
- **Move** the *eyeControl* node to Salty's right close to where the ball is landing.
- Press **Shift w** to key the translation channels of this node.



Eyes following the ball

- Go to frame **440**.
- Select the *eyeControl* group using its selection handle.
- **Move** the *eyeControl* node out in front of Salty near the ground since Salty doesn't want to make eye contact with the audience.
- Press **Shift w** to key the translation channels of this node.



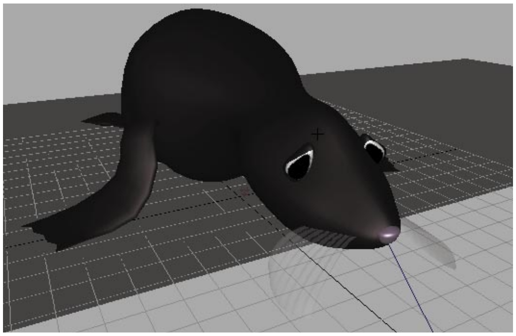
Eyes in front looking down

- Play back the results.

2 Animate the worry channel

You will now animate the worry channel to control the shape of Salty's eyes over time.

- Go to frame **5.**
- In the Channel box, change the **Worry** channel to **0**.
- Click on this channel and use the right mouse button to select **Key Selected** from the pop-up menu.
- Go to frame **190**.
- Again select **Key Selected** to keep the **Worry** set to **0**.
- Go to frame **300**.
- Set **Worry** to **5**.
- Select **Key Selected**.
- Go to frame **440**.
- Set **Worry** to **10**.
- Select **Key Selected**.



Salty worried about dropping the ball

Note: The other two eye attributes will not be animated right now. Blinking will be animated in Lesson 23. The pupil can be animated later as you see fit.

3 Save your work

Making the whiskers flop

With Maya, you can use soft body dynamics to animate Salty's whiskers. Soft body dynamics lets you use forces on an actual surface to get realistic secondary motion.

A soft body is actually a surface that is defined by particles. These particles can then be animated using dynamic forces, and the surface deforms as required.

1 Create the soft body

The whisker surface will be turned into a soft body. The original surface will be hidden and used as a *goal* for the particles. This means that as the whiskers move around using dynamics, they always try to maintain their original shape.

- Go to frame **1**.
- Go to the **Dynamics** menu set.

- Select the *whiskers* surface.
- Select **Bodies** → **Create Soft Body** - ❐ and set the following options:

  Create Options to **Duplicate, Make Copy Soft**

  Hide Non-Soft Object to **On**;

  Make Non-Soft a Goal to **On;**

  Weight to **1**.
- Click **Create** then **Close**.

2 Open the Component Editor

By setting a weight of 1, the soft body will always attempt to match the original whisker surface.

You will now edit the individual particle weights to make the edges of the whisker more free.

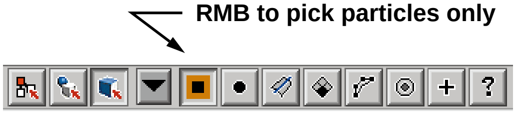
- Select the soft body *whiskers*.
- Open the Attribute Editor.
- In the Attribute Editor, click on the *CopyOfwhiskersParticlesShape* node.
- Open the **Per Particle (Array) Attributes** section.
- With the right mouse button, click on the **goalPP** field and select **Component Editor...** from the pop-up menu.

  This window lets you set the weights for the particles on a per particle basis.

3 Go into select particles mode

To edit the particles, you have to be able to pick them. They can be picked using component mode.

- Press **F8** to go into component mode.
- Click on the **Points** mask button to turn it off.
- Click on the **Points** mask button with the right mouse button and choose **Particles**.

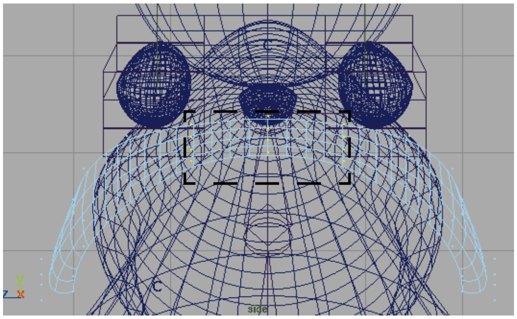


RMB to pick particles only

Selection masks

4 Set goal weights for the points

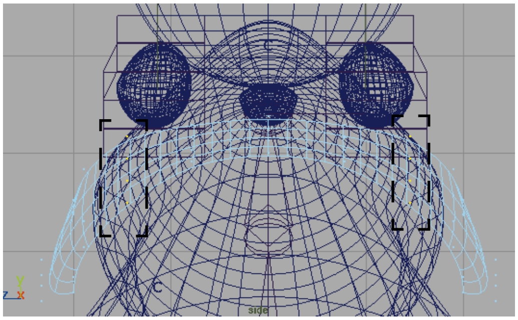
- Set up a Side view panel and dolly in to see the whiskers.
- **Click-drag** a selection box around the three middle rows of particles.
- In the Component Editor, press **Load Components**.
- Click on the *Particles* tab.
- Set the **goalPP** to **1.0** for the selected particles.



Center particle selection

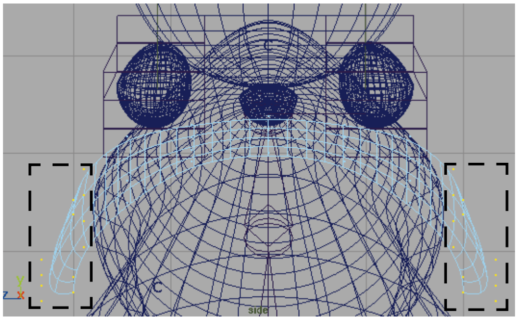
- Select the next two rows of particles on both sides of the whiskers.

- In the Component Editor, press **Load Components**.
- Set the **goalPP** to **0.75**.



Middle particle selection

- Select the outside rows of particles on both sides of the whiskers.
- In the Component Editor, press **Load Components**.
- Set the **goalPP** to **0.50**.



Edge particle selection

5 Link gravity to the whisker soft body

To achieve a realistic effect with the soft body, it is important that it is affected by gravity.

- Press **F8** to go back to object mode.
- Select the *copyOfWhiskers* node.
- Select **Fields** → **Create Gravity.**
- The whiskers are now connected to the gravity field. You can verify this by opening the Dynamics Relationship Editor.

6 Add a spring to the soft body

To make the whiskers bounce, the soft body needs the help of some springs.

- Select *copyOfWhiskers*.
- Select **Bodies** → **Create Springs** - ❐ and set the following options:

  Under **Spring Methods** set:

  Creation Method to **Min/Max**;

  Min Distance to **0**;

  Max Distance to **1**.
- Under **Spring Attributes** set:

  Stiffness to **10.00**.
- Click **Create**.

7 Set up cache for whiskers

- Select *copyOfWhiskersParticle*. You may want to use the Hypergraph to help get the exact node.
- In the Channel box, under the softObjectParticles node, set **Cache data** to **On**.
- Select **Solvers** → **Scene Caching** → **Enable**.
- Play back the results

  This will create a cache file so that you can scrub in the time slider and playback more quickly in subsequent playbacks.

8 Save your work

You will use this scene in your next lesson.

Playblasting the animation

You will now Playblast the scene to test the results. Playblast gives you fast access to your animated sequence and lets you quickly evaluate the results using a Movie Player movie.

1 Set up the Perspective view

- From the Perspective panel, select **Show** → **None** to turn all object types off. Next, select **Show** → **Surfaces** and **Show** → **Meshes** to turn these types on.

  This gets rid of any icons that may be cluttering up the view.

2 Playblast the animation

- Select **Window** → **Playblast** - ❐ and set the following:

  View to **On**;

  Show Ornaments to **Off**;

  Display size to **From Render Globals**;

  Save to file to **Off**.

- Click the **Save** button.
- Select **Window** → **Playblast**.

3 Play back the movie

- Use the Movie Player controls to play back the sequence.

Conclusion

In this lesson you have taken advantage of the set-up time you put into preparing Salty. The cluster handle helped simplify the animation of Salty's neck and the eye control node made it much easier to control various aspects of these important parts of your character.

In the next lesson, you will build a stage for Salty to perform on and then set up the scene's lighting.

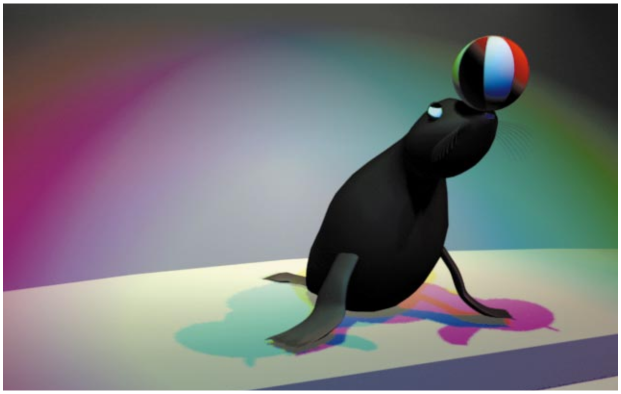
You may be wondering why you didn't set keys on Salty's blinking – you will do this by building a special command in Lesson 23, using Maya's scripting language.

21 Building the Set

In this lesson, you will build Salty's stage which is sitting at the edge of a swimming pool. This stage will be created using: a back surface as a wall, the polygonal deck surface created earlier, two polygonal surfaces which will act as the side and bottom of a pool, and a NURBS surface that will act as the water's surface.

Afterwards, you will place lights into the scene to create some mood and atmosphere. Several lights will be used to give the stage the look of a performance.

These surfaces will be built using very simple geometry that will be enhanced in later lessons when you build and assign shading groups to the various parts.



Salty's stage

In this lesson you will learn the following:

- How to draw a spline curve
- How to create a lofted surface
- How to use curve snapping
- How to place multiple lights into a scene
- How to add color to the lights

Initial set-up

To start this lesson, you will continue with the file from the last lesson.

1 Create a layer for the set

To help manage the scene, you will set up a second layer for the new objects and the deck surface.

- Open the Layer Editor by selecting **Window → Layer Editor**.
- Create a new layer and call it *stage*.

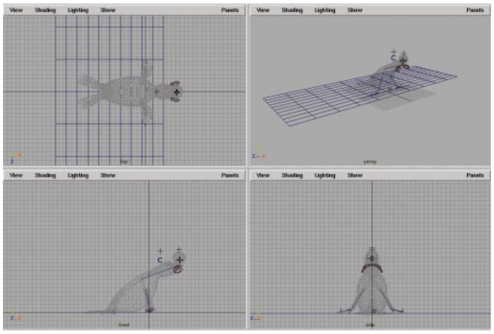
2 Template the seal layer

- Click on the *Salty* layer name.
- Click the **Template** button to template everything on the layer.

3 Set up your view panels

- Select **Panels → Saved Layouts → Four View**.

  Make sure that all of the view panels are in wireframe mode. This will make it easier to see the wire lines as you build the stage.



Four view panels

Creating the pool

The first part of the set is the pool which will be built using polygonal surfaces that are the same size as the deck surface.

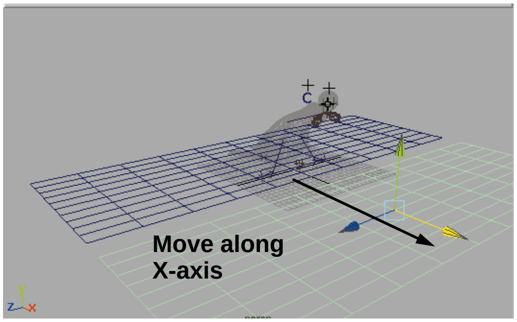
1 Create a new polygonal surface

You will create the side and bottom of the pool using a polygonal surface that has the same size as the deck surface.

- Go to the **Modeling** menu set.
- Select **Create → Polygon Primitives → Plane**.
- Rename the surface *poolSide.*
- Click on the *polyPlane* input node and set the floor as follows:

  Width to **25**;

  Height to **75**.
- **Move** the surface along the X-axis so that it is positioned in front of the deck surface.

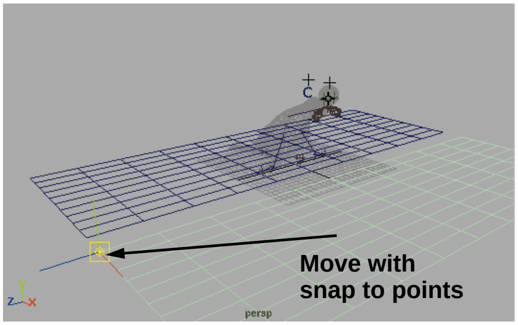


New surface

2 Position the pivot point

You will now move the surface's pivot point from the center to one of the corners. This will make it easier to allow the side of the *poolSide* surface to share a common edge with the *deck* surface.

- Press the **Insert** key on the keyboard.
- Hold down the **v** key to activate the **Snap to points** function. This will let you snap to the edge of the surface.
- Drag the center of the pivot manipulator to the far corner of the poly surface.

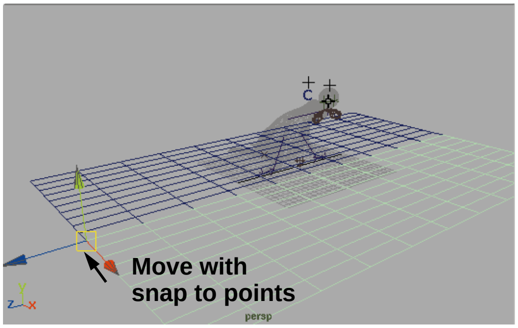


Relocated pivot point for surface

3 Position the pool surfaces

The pivot point's new position allows you to snap the two surface edges together.

- Press the **Insert** key again to return to the move manipulator.
- Hold down the **v** key.
- **Move** the *poolSide* surface to the front edge of the *deck* surface.



Relocated surface

4 Duplicate and rotate the surfaces

The third surface is created by duplicating the second surface. This will maintain the repositioned pivot point. The new surface is then rotated about the common edge.

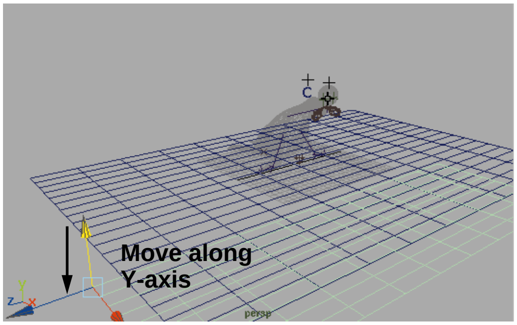
- Select **Edit** → **Duplicate** - ❐ to open the option window and set the following:

  Duplicate Upstream Graph to **On**.

- Click **Duplicate** and **Close**.

  The third surface is created with a duplicated *polyPlane* input node.

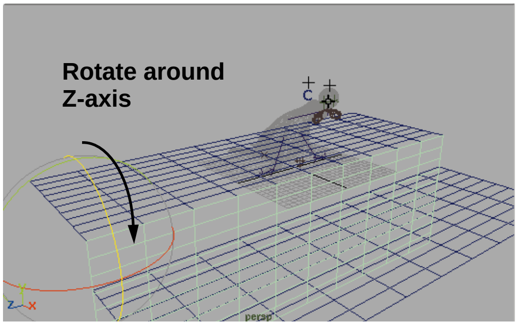
- Rename the surface *poolBottom*.
- **Move** the duplicate surface down to form the pool bottom.



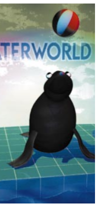
Moved surface

- **Select** the *poolSide* surface.
- **Rotate** the surface -90 degrees around the Z-axis.

The pool and deck are now complete.



Rotated surface



Tip: By keeping the surfaces the same size and proportion, one shading group can later be applied to all three surfaces without affecting the detail of the texture maps. In this case, you should let the surfaces overlap to maintain the same size and proportion.

5 Create the water surface

To simulate the effect of a pool filled with water, the surface of the water will be created as a NURBS surface which will later be textured to appear wavy.

- Select **Create** → **NURBS Primitives** → **Plane**.
- Rename the surface *water*.
- Click on the *makeNurbsPlane* input node and set the floor as follows:

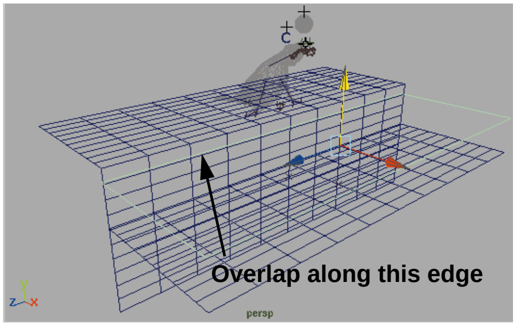
  Width to **25**;

  Length Ratio to **3**;

  U Patches to **1**;

  V Patches to **1**.
- **Move** the surface about 1 unit below the *deck* surface.
- **Move** it in front of the *poolSide* surface as shown below.

  It is not necessary for the edge of the water surface to be snapped to the side of the pool. If the surfaces intersect each other, the rendering process will define the line of intersection.



Water surface in position

Note: A displacement map will be assigned to this surface to create the waves and, by intersecting them, it ensures that the water remains tight to the side of the pool – instead of creating a gap.

Creating the back wall

You will now create a curved vertical wall as the backdrop to the scene. This wall will be created by lofting two curves to create a NURBS surface.

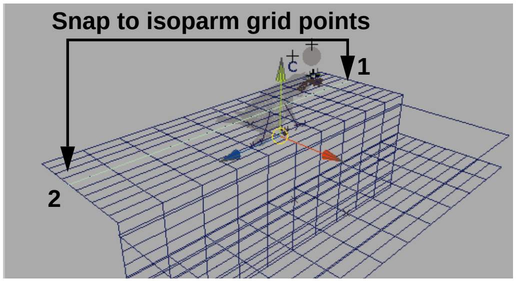
In the last lesson, you learned how to sculpt a NURBS surface by editing the control vertices. In this situation, you will focus on creating a curve with the desired shape which you can then use to create the surface.

1 Draw a curve

To form the base line of the wall, a straight single span curve is positioned to stretch between the opposite edges of the deck surface. For this purpose, an edit point curve will be used.

- Select **Create → EP Curve Tool**.
- Hold down the **v** key.
- Snap the first edit point to one of the *deck* surface's isoparm grid points.
- Snap the second point to the opposite side of the surface.

  Because the curve is coincident with the center line of the surface, it cannot be seen while it is being created.
- Press the **Enter** key to finish the curve construction.



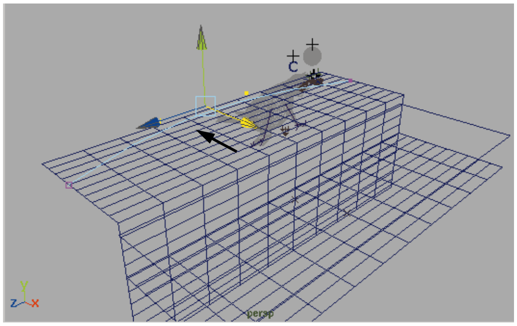
Single span curve snapped into position

Note: It is important that you only click once at each location, or else two edit points will be placed at the same location; the curve will extend beyond the points and double back on itself.

2 Select and move the center CVs

By moving the curve's two middle CVs, you create a symmetrical and uniform curve that stretches between the edges of the deck surface. To do this, the curve's CVs must first be displayed.

- Press **F8** to change the selection mask to **Select by component type**.
- **Select** the two CVs close to the middle of the curve.
- **Move** the CVs back away from the seal.



CVs moved to shape the curve

3 Duplicate and move the curve

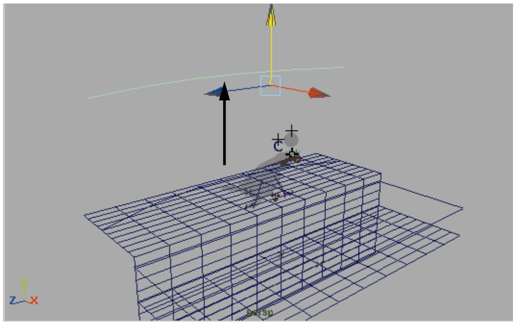
The top of the wall is defined by a copy of the curve.

- Press **F8** to change the selection mask back to **Select by object type**.
- **Select** the curve.
- Select **Edit** → **Duplicate** - ❐ to open the option window and set the following:

  Duplicate Upstream Graph to **Off**.

- Click **Duplicate** and **Close**.
- **Move** the duplicate up along the Y-axis.

  The curve should be positioned at the desired location for the top edge of the back wall.



New position of second curve

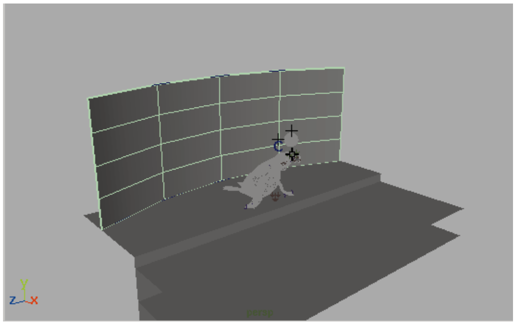
4 Create a loft surface

A single span NURBS surface is created that connects between the two curves.

- **Select** the two curves.
- Select **Surfaces** → **Loft**.

  A surface is created between the two curves.

- Press the **3** key to increase the surface smoothness and the **5** key to turn on smooth shading.



Finished back wall

LIGHTING THE SET

Now that the set is ready, you will add lights. Lighting is an important part of any scene since it adds mood and atmosphere.

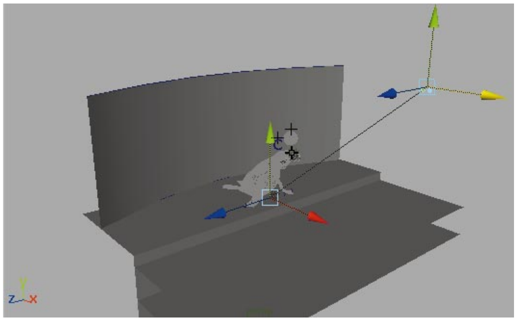
Creating spot lights

In this scene, you will illuminate Salty with three main spot lights that will be colored green and blue. When all these lights combine you will get white light with fringes of color at the edge.

1 Create and position a spot light

You will create a spot light in the scene then reposition it using the show manipulator tool. You will then use the *look through* function to more accurately position the light.

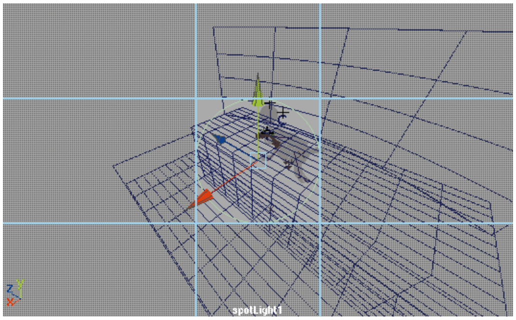
- Go to the **Rendering** menu set.
- Select **Lights** → **Create Spot Light**.
- Select the **Show Manipulator** tool.
- Position the *look at* and *eye* points of the light so that the light is pointing down at Salty from the upper left.



New position of second curve

- From one of the orthographic view's **Panel** menu, select **Look Through Selected**.

The panel view changes to a Perspective view that looks through the spot light.

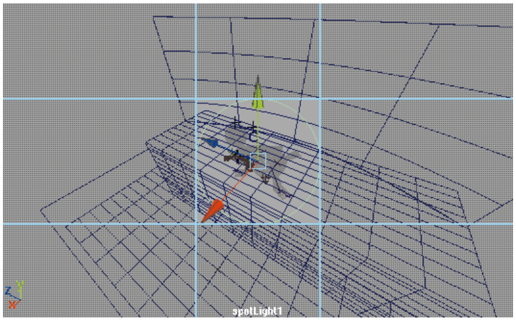


View looking through the spot light

- Dolly, tumble, and track the view in the spot light view panel.

The position and direction of the spot light changes to that of the changed view.

- Change the position of the view so that it is centered on the seal from above.



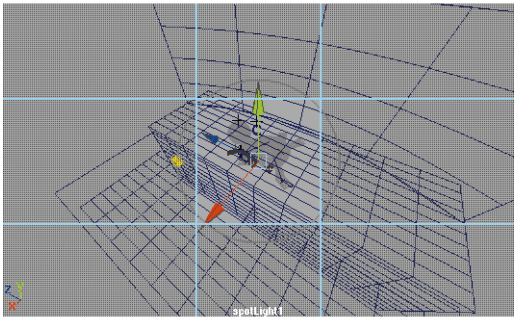
View through the repositioned spot light

2 Change the angle of the spot light

The cone angle of the spot light is changed using the light's multilayered manipulators.

- Click twice on the small blue manipulator button or select **Display → Camera/Light Manipulator → Cone Angle**.
- The circular outline in the spot light view represents the outside extent of the spot light's cone.
- Click-drag on the blue dot of the manipulator to change the cone angle.

  The circular outline of the manipulator should slightly overlap both the *wall* and the *water* surfaces.



Modified cone angle of the spot light

3 Change the penumbra of the spot light

The spot light is created with a hard outside edge to the cone of light. The light can be given a softer edge by adjusting the penumbra.

- Click once on the small blue manipulator button or select **Display → Camera/Light Manipulator → Penumbra**.
- Click-drag the manipulator in the spot light view to increase the angle of the penumbra slightly.

  This softens the edge of the spot light.

4 Change the spot light color

You will now give the spot light a color to create more dramatic lighting, and to differentiate it from the other light sources that will be added later to the scene.

- Press **Ctrl a** to open the Attribute Editor.
- Change the **Color** to a full, saturated red.

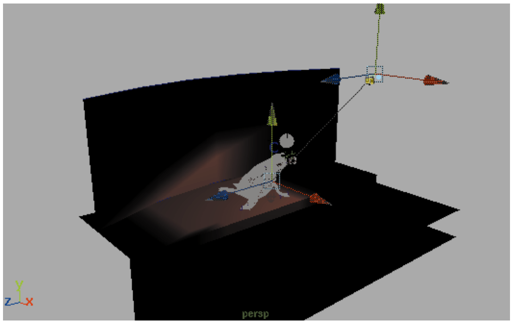
5 Use smooth shading and default lighting

In order to see the approximate affect of the spot light, you can use Maya's smooth shading and lighting in the hardware rendering of the views.

If you do not have hardware texture RAM, then skip to the next step and you can test render the scene after the lights have been placed.

- Press the **7** key to turn on **Use All Lights** in the Perspective view panel.
- Press the **5** key then the **7** key in the spot light view panel to turn on smooth shading and the lights.

  You can now see the lighting, but it seems a little bit broken and fuzzy.



Smooth shading and default lighting

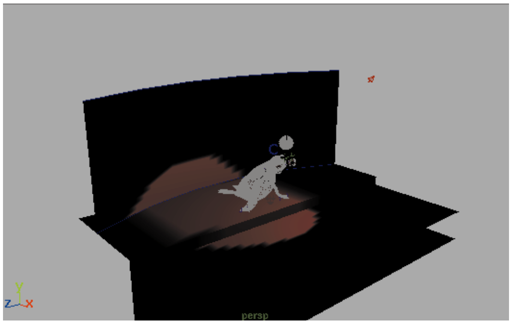
6 Increase the NURBS smoothing

The curved wall is displayed as a flat plane and the lighting that is not rendering the affect of the spot light very well. This is because of the low display subdivisions of the NURBS surface.

- **Select** the *wall* surface.
- Select **Display → NURBS Smoothness → Custom - ❏.**
- In the **Shaded** section of the option window:

  Set **Surface Div per Span** to **32**.
- Click **Apply** and **Close**.

  The light areas of the surface are now more defined.
- **Select** the *water* surface.
- Select **Display → NURBS Smoothness → Custom.**



Increased smoothness

Tip: The higher the surface shading divisions are set, the smoother the hardware lighting will appear – but increased division settings will slow down the interaction of the model views.

7 Increase the polygon smoothing

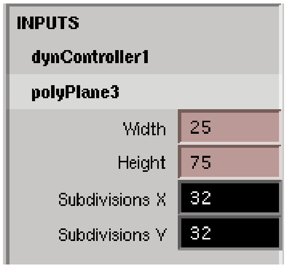
The detail of the hardware lighting conditions for polygon surfaces is determined by the size of the polygons – the smaller the polygons, the more detailed the lighting conditions.

In this case, the size of the polygons is determined by the number of polygon subdivisions which is set in the input shape node.

- **Select** the deck, *poolSide* and *poolBottom* polygon surfaces.
- In the Channel box, click on the *polyPlane* node under the **Inputs** section.
- Click into the *Subdivision X* field then press the **Shift** key and click into the *Subdivision Y* field.

The first value highlights in white while the second highlights in black, allowing both to be changed at once.

- Enter a value of **32**.



INPUTS

dynController1

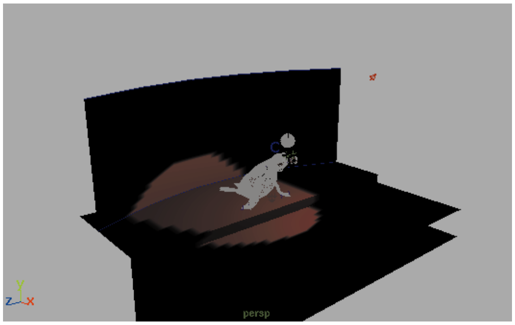
polyPlane3

Width	25
Height	75
Subdivisions X	32
Subdivisions Y	32

Entering values in the Channel box

The subdivisions of all three surfaces are increased. When a series of objects or nodes are selected, the changes that are made in the Channel box affect the particular attribute on all of the selected nodes.

- Deselect the surfaces to see the increased detail of the lighting.

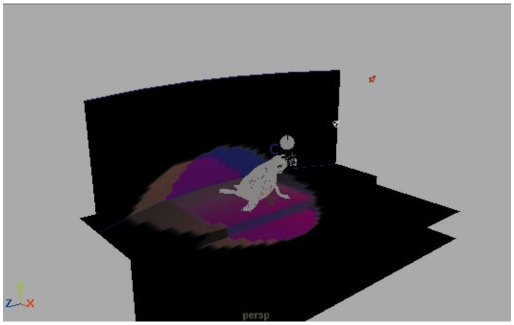


Increased polygon subdivisions

By increasing the smoothness of the surface display, you can see the lighting more accurately.

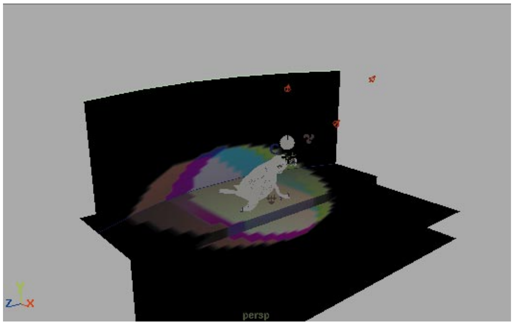
8 Duplicate and position the spot lights

- **Select** and **Duplicate** the spot light.
- In the spot light view panel, select **Panel → Look Through Selected**.
- Tumble the spot light view to a position looking from the right side.
- Change the spot light **Color** to full saturated blue.



Three different colored spot lights

- Repeat the procedures above to create a green spot light positioned to the left.

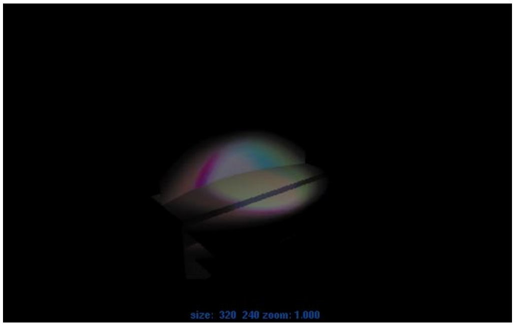


Three different colored spot lights

9 Test render the scene

If you do not have texture RAM, then you should test render the scene to see the colored lights.

- In one of the orthographic panels, select **Panels → Panel → Render View**.
- From this panel, select **Render → Render → persp**.



Software rendering of set

Note: Salty does not render because she is templated.

Creating background lighting

With the main spot lights established, it is necessary to light the background wall of the stage. You will also create additional ambient lighting.

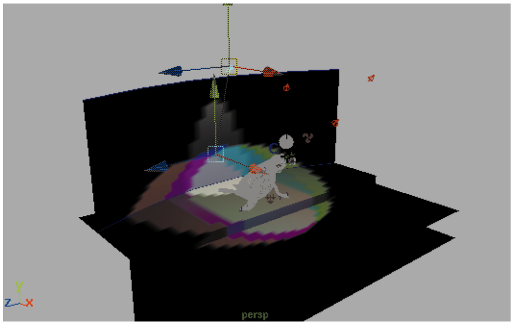
1 Create and position a spot light

A series of spot lights will be used to illuminate the back wall. The lights should focus on a centralized area of the wall where a logo will be later assigned.

- Select **Lights → Create Spot Light**.

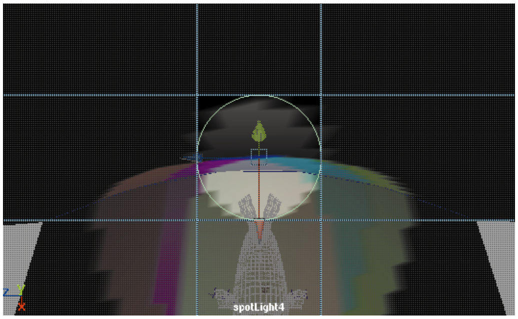
The light is positioned at the origin.

- Select **Show Manipulators**.
- Position the manipulators so that the light is shining down on the center of the back wall as shown.



Hardware rendering of the light position

- If you cannot see the lights with hardware rendering, use the **Look Through Selected** to position the light.



View looking through the spot light

Note: If hardware rendering is on, you may notice a slowing of the interactivity while positioning the lights, due to the increased display smoothness.

2 Set spot light parameters

To contrast the dynamic colored spot lights that focus on the seal, several identical spot lights illuminate the background with a warm soft light that washes the wall.

- In the Channel box, select the new light's *spotLightShape* node and set the following:

  RGB to **1 1 0.75**;

  Intensity to **0.5**;

  Cone Angle to **60**;

  Penumbra Angle to **20**;

  Dropoff to **5**.

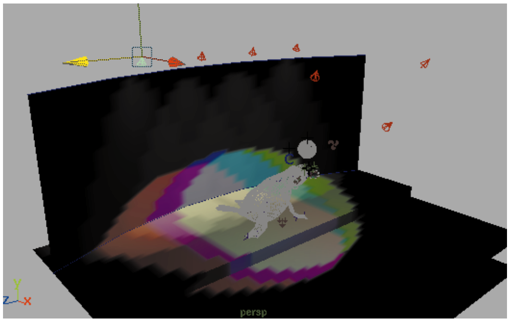
  The Dropoff setting causes the light to lose intensity as the distance from the source increases. The RGB settings define a pale yellow color for the light.

3 Duplicate and position the lights

Once the settings are established for the first light, it can be duplicated to create an even pattern of lights across the back wall.

- **Move** the spot light to one side of the Z-axis.
- **Duplicate** the spot light.
- **Move** the new light to the other side of the axis.
- **Duplicate** and **Move** two more spot lights.

  The four spot lights should be equally spaced and centered around the axis line.



Hardware rendering of four lights

4 Adjust all of the spot lights

In situations where you have multiple lights which you want to stay the same, the Channel box can be used to modify any of the attributes.

- Select all four of the background spot lights.

  In the Channel box, you can see that more than one node is selected because the node name is followed by three dots.
- In the Channel box, click on the top light's *spotLightShape* node.
- Change the **Cone Angle** to **70**.

  The cone angles are now set for all four of the selected lights.

Tip: The attributes for the node that was selected last will be displayed in the Channel box. If different node types are selected, the changes that are made to the attributes will only affect those nodes that have the attributes.

5 Add an ambient light

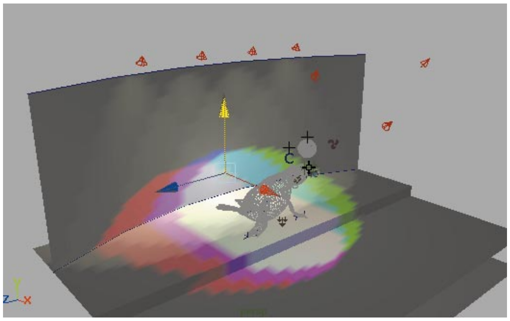
Ambient lighting is needed to fill in the shadows and areas of the scene that are not directly lit with the spot lights. Even though there are a lot of lights in the scene, the light from a spot light will not "bounce" but will only illuminate what it is directly shining on. Ambient light is good for filling in the dark areas.

- From the **Rendering** menu set, select **Lights** → **Create Ambient Light**.

  If you have hardware lighting turned on, you will see that this light significantly increases the illumination in the scene.

- **Move** the light behind Salty and slightly above the deck surface.

  The light in this location will approximate the light that is expected to bounce off the deck.



Ambient light in position

- In the Channel box set:

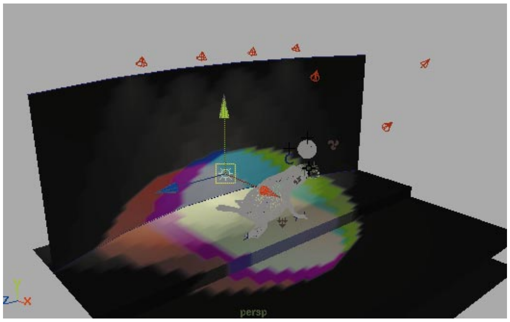
  Intensity to **0.2**.

- In the Attribute Editor set:

  Color to a pale turquoise;

  Ambient Shade to **0.75**.

  The Ambient Shade attribute affects the amount that the ambient light illuminates surfaces that are facing away from the source.



Position of the ambient light source

Animating the camera

To preview the animation, you will now add a new camera and animate its position. When working in 3D space, an animated camera can help accentuate the depth of a scene.

1 Untemplate the Salty layer

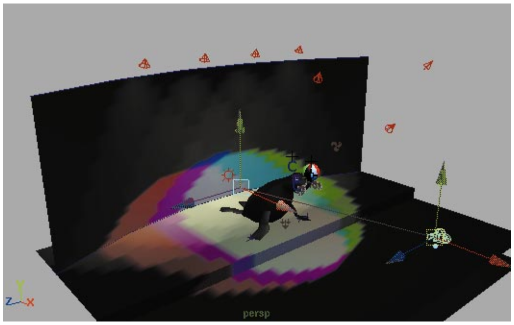
- Click with **RMB** over the *Salty* layer button and select Standard from the pop-up menu.

2 Create a new camera

It is a good idea to always create a new camera for your animations. The default Perspective view should be saved for modeling, and set up while the new camera is used for the final take.

- Select **Create** → **Camera**.
- Select the **Show Manipulator** tool.

  Position the look at point of the camera just behind Salty and the eye point of the camera in front.



Position of new camera

- From the Perspective view panel, select **Panels → Perspective → camera1**.

  Don't worry if you don't like the resulting view. It's difficult to properly position a camera using the manipulators. You can now fix the view using the normal view tools.

3 Set focal length

- In the Channel box, set the **Focal Length** to **30**.

  This widens the view angle to show more of the scene.

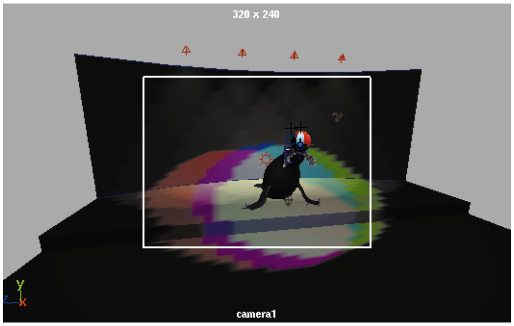
4 Position the camera

- From the Perspective panel, select **View → Camera Settings → Resolution Gate**.

  The view expands and a line is drawn around the area that will be rendered. This allows you to see the scene and animation outside of the rendered area.

- Dolly, track and tumble in this view to get a good view of Salty.

You should see the water surface in the foreground and the illuminated area of the back wall.



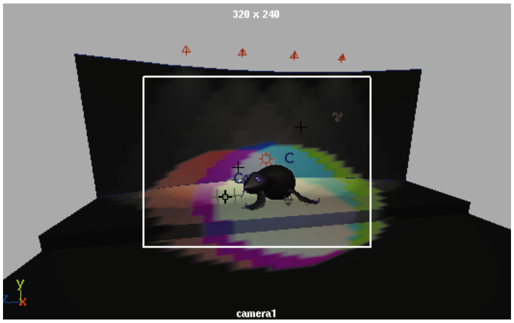
Scene with resolution gate on

5 Keyframe the camera position

- Go to frame **340**.

  This should be the point where Salty is just finished watching the ball bounce away.

- From the *camera1* panel, select **View → Select Camera**.
- Press the **s** key to set a key at this position.



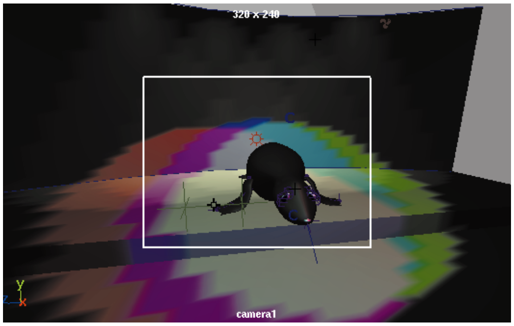
Scene at frame 340

6 Keyframe a second position

- Go to frame **400**.

This should be the point where Salty is just finished watching the ball bounce away.

- Dolly in for a close-up of Salty and press the **s** key to set a key at this position.



Scene at frame 400

Playblasting the animation

You will again Playblast the scene to test the results. It is always good to do rough tests to make sure that everything is working properly.

1 Set up the Perspective view

To Playblast you do not need to see the resolution gate outline.

- From the Perspective panel, select **View → Camera Settings → No Gate**.
- From the Perspective view, select **Show → None** to turn all object types off then select **Show → Surfaces** and **Show → Meshes** to turn these types on.

2 Playblast the animation

- Select **Window → Playblast... - ❐** and set the following:

  View to **Off**;

  Show Ornaments to **Off**;

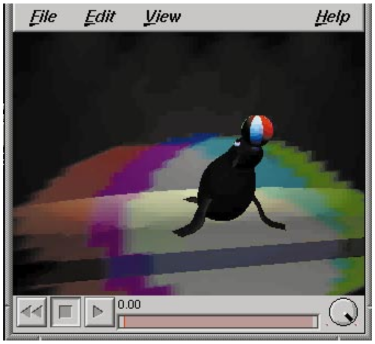
  Display size to **From Render Globals**;

  Save to file to **Off**.

- Click the **Save** button.
- Select **Window → Playblast...**

3 Playback the movie

- Use the Movie Player controls to play back the sequence.



MoviePlayer view of Salty

4 Save your work

If you like, save your work.

- Select **File → Save Scene As...** and save your work as *Salty5*.

Conclusion

Your scene is now built and lighting has been applied. Now you can build shading groups for the various surfaces to create a more interesting environment for Salty to bounce her ball.

TERWORLD

22 Texture Mapping

In this lesson, you will add different kinds of texture maps to the stage, then prepare the file for rendering. The stage's simple surfaces will become more visually complex using textured shading groups.

The tiled deck and pool surfaces will be textured using a grid *bump map*, giving an appearance of being raised. The water surface will be created using a *displacement map* that actually raises the surface. The water's texture will include animated attributes to give the water a more realistic appearance.

WATERWORLD

The textured set

In this lesson you will learn the following:

- How to use a bump map texture
- How to build and animate a displacement shader
- How to build a layered shader
- How to set up a textured light
- How to set up a raytrace rendering
- How to render the final animation

Initial set-up

To start this lesson, you will continue with the scene file from the last lesson.

1 Set up your panel arrangement

- Select **Panels** → **Saved Layouts** → **Hypershade/Render/Persp.**
- In the Perspective view panel, select **Panels** → **Perspective** → **camera1**.

Creating the deck shading

The floor and the side of the pool will be textured using a series of grid textures that give it color and material qualities.

1 Create and assign a phong material

To produce a shiny, reflective surface for the floor and side of the pool, a phong material node will be used.

- In the Hypershade panel, **Create** a new **Phong** material.
- Rename the nodes *tileSG* and *tilesPhong*.
- **Assign** the *tileSG* node to the *deck*, *poolSide* and *poolBottom* geometry.

Tip: The nodes can be renamed in the Hypergraph by double-clicking on the swatch with **ctrl** key name bar and entering a new name.

2 Map a grid texture to the color

Instead of using a simple color for the material, a grid texture is used to produce a colored tile pattern on the floor.

- Open the *tilePhong* node's **Attribute Editor**.
- Click on the **Map** button next to **Color**.
- In the Create Render Node window, make sure the **Textures** tab is up.
- From the 2D texture section, select the **Grid** texture.

3 Set the grid color

Once the texture is assigned, the parameters of the texture node can be adjusted. The overall fill color of the tiles is set to one color, while the joints between the tiles is set to be a different color.

In the **Attribute Editor**:

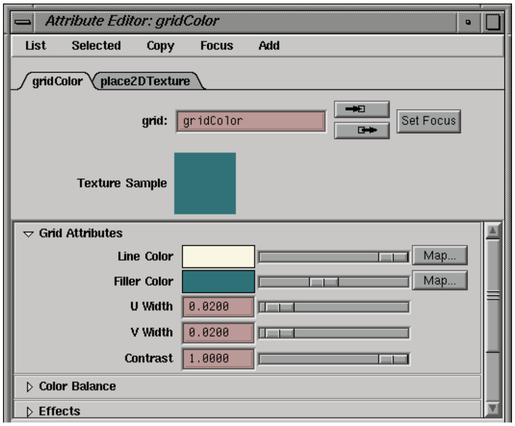
- Rename the node *gridColor* and set the following:

  Line Color to a pale yellow;

  Filler Color to a dark turquoise;

  U Width and **V Width** to **0.02**.

A turquoise grid with thin yellow lines is produced.



grid: gridColor

Line Color

Filler Color

U Width 0.0200

V Width 0.0200

Contrast 1.0000

Settings in Attribute Editor

- Next open the **Effects** section and set the following:

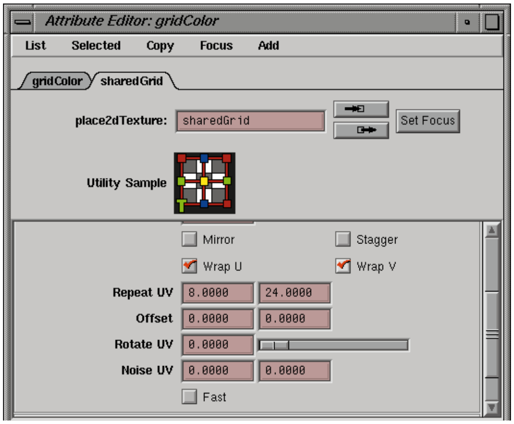
  Filter to **0.5.**

4 Set grid placement

- Click on the *place2dTexture* tab in the Attribute Editor.
- Rename the node *sharedGrid*.

  This node will be shared by other textures later.
- Increase the **Repeat UV** to **8** and **24**.

  This changes the number of grid squares across the entire surface.



place2dTexture: sharedGrid

□ Mirror □ Stagger

☑ Wrap U ☑ Wrap V

Repeat UV 8.0000 24.0000

Offset 0.0000 0.0000

Rotate UV 0.0000

Noise UV 0.0000 0.0000

□ Fast

Attributes of the texture placement

5 Set your Render Globals

Even if you are able to display the scene with hardware texturing, the quality of the texture does not display very well on the surfaces. It is necessary to create a software rendering of the surfaces in order to evaluate the tile pattern.

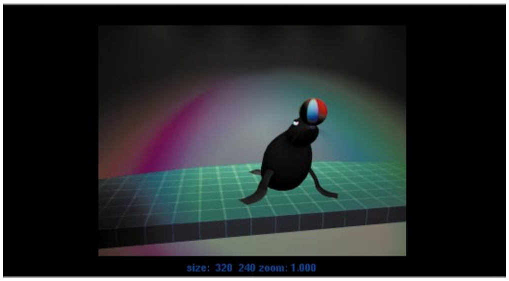
- In the **Render View** panel, select **Options** → **Render Globals...**
- Open the **Anti-Alias Quality** section.
- Set the **Presets** to **Intermediate Quality**.

  Notice that the anti-aliasing attributes change.
- Open the **Resolution** section.
- Set the **Render resolution** to **320x240.**

6 Test render the scene

- In the **Render View** panel, select **Render** → **Render** → **camera1**.
- Once the rendering is finished, press the **Keep Image** button.

  This allows you to keep a series of renderings for comparison purposes so that you can watch the development of the shading group.



Test rendering of the color texture

Note: A scroll bar is placed at the bottom of the panel which you can use to scroll between the different test renderings. Renderings that are kept are not saved onto the hard drive and are only maintained while Maya is running.

Adding a bump map

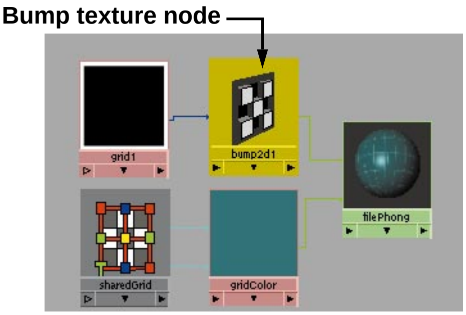
A *bump map* is a special map type that lets you render a surface as if it has some surface relief. The areas of black, gray and white on the bump map create the effect of a raised surface when rendered.

1 Map grid to bump node

A second grid texture is used to produce a bump map that will produce the appearance that the lines between the tiles are recessed. A second texture node is used so that the thickness of the lines can be different than the *gridColor*.

- In the Hypershade, double-click on the *tilePhong* node.
- Click on the **Map** button next to **Bump Mapping**.
- In the **Textures** section of the **Create Render Node** window, turn **With New Texture Placement** to **Off**.
- From the 2D texture section, select **Grid**.

  In the Hypergraph, a *bump2d* node is placed between the texture and material nodes, but no texture placement node is produced.



Multilister view of the shading group

2 Set bump and texture attributes

The depth of the bump is set in the bump node, and the areas that are to be recessed are set in the grid texture node. Any color that is used in the texture will be ignored, because the bump map only deals with the grayscale or *alpha channel* values.

- In the **Attribute Editor** for the *bump2d* node, set the following:

  Bump Depth to **-0.1**.

The dark areas of the texture will appear slightly recessed.

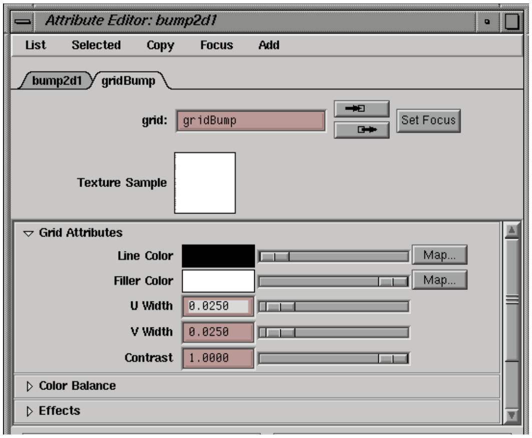
- Click on the new *grid* tab to bring this node forward and set the following:

  Name to *gridBump;*

  Line Color to black;

  Filler Color to white;

  U Width and **V Width** to **0.025**.



grid: gridBump

Line Color

Filler Color

U Width 0.0250

V Width 0.0250

Contrast 1.0000

Bump texture attributes

- Next open the **Effects** section and set the following:

  Filter to **0.5.**

This blurs the texture less than the default setting of 1.0. For most textures, you will generally want a lower filter setting.

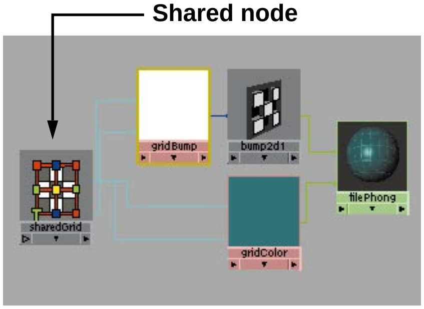
Tip: The line thickness for the bump map is set to be thicker than the color texture so that the bump will occur within the tile color instead of the fill color.

3 Share the texture placement node

Because the color map and bump map need to be positioned identically on the surface, the texture placement node for the color map will be used for the grid map.

- With the middle mouse button, click-drag the *sharedGrid* swatch onto the *gridBump* swatch.

  The 2D texture placement node for the first grid is connected to the second grid.



Hypergraph view of shading group

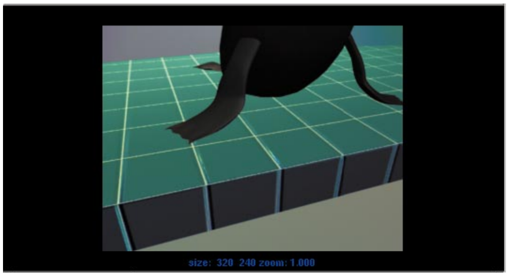
Tip: You could use separate texture placement nodes but you would have to make sure that they both had identical attribute settings. Using the one node simplifies changes that you could make to the number of tiles on the surface.

4 Test render the scene

To see the effect of the bump map, the scene must be rendered.

- Dolly into the Perspective view to get closer to the tiles.
- In the **Render View** panel, select **Render → Redo Previous Render**.
- Once the rendering is finished, use the scroll bars at the bottom of the panel to compare this rendering with the previous one.

  If the view or lighting conditions have not been modified, it is easy to evaluate the material before and after the change.
- Press the **Keep Image** button to keep this rendering.



Rendering of bump texture

Note: If you do not select to keep the latest rendering, it will be replaced the next time the scene is rendered.

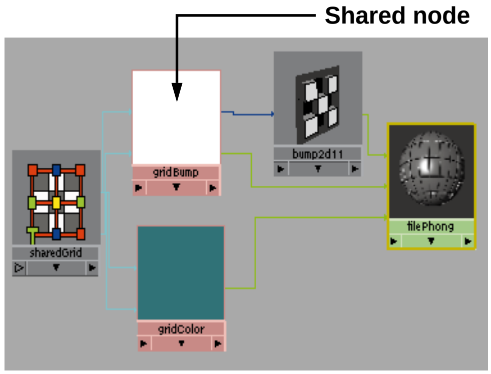
Refining the floor material

You will now reassign the bump map as a specularity map. You will then preview the results using a *raytrace* rendering. Raytracing lets you see reflections and refractions.

1 Connect node to shading group

You will now connect the *gridBump* node to the material node as a specularity map. This texture can now play two roles.

- In the Hypershade, click-drag with middle mouse button the *gridBump* swatch onto *tilePhong*.
- Select **Specular Color** from the pop-up menu.



Hypergraph view of shading group connections

2 Test render the scene

In order to see the subtleties of the bump texture and added effects of the shading group, you can produce a test rendering that uses raytracing.

- From the Render View panel, select **Options → Render Globals...**

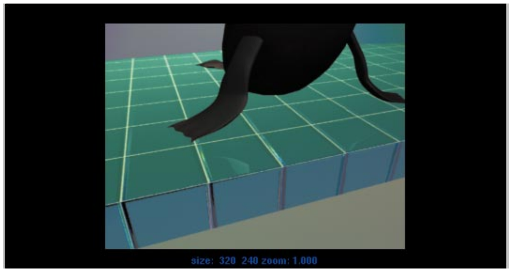
- Open the **Raytracing Quality** section and set the following:

  Raytracing to **On**;

  Reflections to **1**;

  Refractions to **1**;

  These settings only allow 1 level of reflections, and refractions. This will keep rendering times faster for your tests.
- In the Render View panel, select **Render → Redo Previous Render**.



Rendered floor with raytracing

The water shading group

The water is going to be textured using a color texture that is also mapped as a displacement map. This means that the surface will raise and lower based on the value of the map texture. Compared to bump mapping, a displacement map is slower but achieves more realistic results.

1 Create and assign a Blinn material

- **Create** a new **Blinn** material node with shading group.
- Rename the nodes *waterSG* and *waterBlinn.*
- Assign *waterSG* to the *water* surface.

2 Edit transparency, specular and raytrace attributes

- Open the *waterBlinn* node in the Attribute Editor and set the following:

  Transparency to about **0.3.**

Tip: Open the color swatch and set **Value** to about **0.3**.

In the **Specular Shading** section:

Eccentricity to about **0.06**;

Specular Roll Off to over **0.9**;

Specular Color Value to **0.7**;

Reflectivity to **0.4**.

- In the **Raytrace Options** section:

  Refractions to **On**;

  Refractive Index to **1.33**;

  Refraction Limit to **1**;

  Reflection Limit to **1**.

3 Create a water color map

- In the Hypershade, select **Create → Create Render Node...**
- In the **Textures** section of the Create Render Node window, turn on **With New Texture Placement**.
- In the 2D texture section click on the **Water** texture.
- Rename this node *waterColor*.
- In the **Water Attributes** section of the Attribute Editor, set the following:

  Number of Waves to **20**;

  Wave Amplitude to **0.075**;

  Smoothness to **1.0**.

- In the **Color Balance** section, set:

  Color Gain to a bright turquoise-blue;

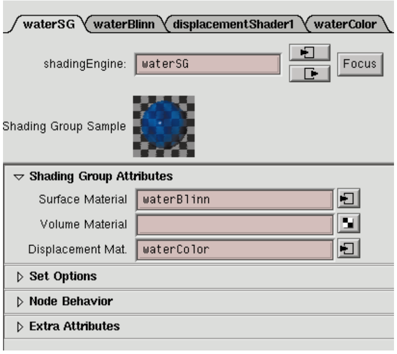
  Color Offset to a dark saturated blue;

4 Connect the map to the shading group

- Using the Hypershade, click-drag with middle mouse button the *waterColor* texture node onto the *waterBlinn* material node. select Color from the pop-up menu.

5 Create a displacement node

- Open the *waterSG* node's Attribute Editor.
- Drag the *waterColor* texture node onto the **Displacement Material** attribute.



shadingEngine: waterSG

Surface Material waterBlinn

Volume Material

Displacement Mat. waterColor

Displacement in shading group

- View the shading group in the Hypershade.

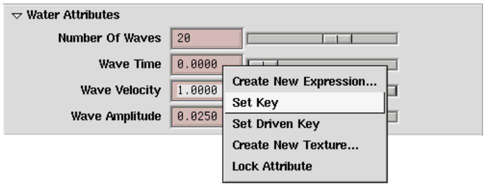
  Notice that the shared texture node is connected to the material node and to a new displacement shader node. This node is similar to a bump node except that it connects directly to the shading group node and not to the material node.

Note: A displacement map moves the actual geometry when rendering, whereas a bump map only gives the rendered appearance of bumps in the material surface.

6 Animate the waves

Since texture nodes are just like other nodes in Maya, their attributes can be keyed. You are now going to key some of the wave attributes so that the water moves during the animation.

- Set the Time slider to frame **1**.
- Open the *waterColor* node's Attribute Editor.
- With the right mouse button, click and hold on the **Wave Time** attribute.
- From the pop-up menu select **Set Key**.



Water Attributes
Number Of Waves 20
Wave Time 0.0000
Wave Velocity 1.0000
Wave Amplitude 0.0250

Setting a key on a texture node attribute

- Change the **Wave Frequency** to **5.0**.
- Select **Set Key** from the pop-up menu for the **Wave Frequency** attribute.

  The attribute fields is colored to indicate that keys have been set for these attributes.
- Change the current frame to the last frame of the animation.
- Change the **Wave Time** to **2**.
- Select **Set Key**.
- Change the **Wave Frequency** to **6.0**.
- Select **Set Key**.

7 Change the direction of the waves

- Open the texture placement node for the *waterColor* node in the Attribute Editor and set the following:

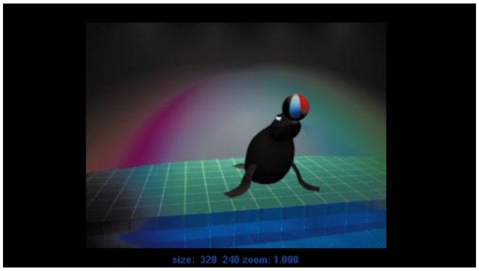
  Rotate Frame to **90**.

8 Test render the water material

- Go to first frame button in the Time slider.

  This should pop your camera back to its initial position.
- In the Render View panel, select **Render** → **Render** → **Camera1**.

  Oops! there doesn't seem to be very many waves.



Raytraced rendering of the water

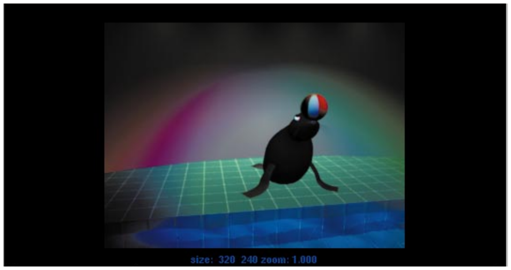
9 Increase the surface tessellation

- Select the water surface.
- Open the Attribute editor.
- Open the **Tessellation** section, and then the **Explicit Tessellation Attributes** section**.**
- Set **Explicit Tessellation Attributes** to **On**.
- In **Primary Tessellation Attributes** set the following:

  Set **Number U** to **50**;

  Set **Number V** to **50**.
- In the Render View panel, select **Redo Previous Render**.

  Notice that the detail of the waves has increased dramatically.



Rendering with increased surface tessellation

Note: Increasing the tessellation of the surface can greatly increase the rendering time but is needed to achieve the detail.

LAYERED SHADERS

A unique material node type in Maya is the *layered shader node*. This material node lets you layer other material nodes together to create more sophisticated effects. The layered shader can be fed by other material nodes and can therefore combine different material effects in one shading group.

The wall shading group

The back wall is going to be designed to look like a plaster wall that is peeling to reveal wood panels below. A layered shading group will be used to achieve this effect.

1 Create a layered shading group

- Create a **Layered Shader** material node with a shading group node.
- Rename the nodes *wallSG* and *wallLayered*.
- **Assign** the shading group to the back wall surface.

2 Create the paint material

A material node is created for the paint and logo that will be applied to the back wall. The material qualities should be set to represent a painted surface with an image texture mapped as a painted logo or sign. Because this node will be included within the layered shader, it does not need to be created with a shading group node.

- In the Hypershade, select **Create** → **Create Render Node...**

- In the **Materials** tab of the Create Render Node window:

  Turn off the **With Shading Group** option.

  Create a **Blinn** material node.
- Open the Attribute Editor for the new Blinn node.
- Rename the new node *paintBlinn*.
- In the **Specular Shading** section, set:

  Eccentricity to **0.4**;

  Specular Roll Off to **0.9**;

  The **Value** of the **Specular Color** to **0.8**;

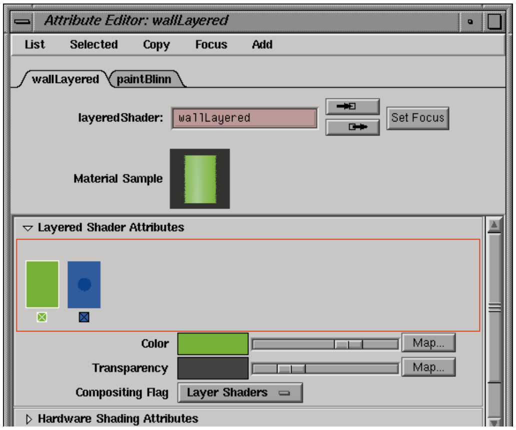
  Reflectivity to **0.1**.

3 Add the material to the layered shader

- Open the *wallLayered* node's Attribute Editor.

  The **Layered Shader Attributes** section contains a green default swatch. This is the default layer.
- From the Hypershade, drag the *paintBlinn* node onto the *wallLayered* Layered Shader Attributes section in the Attribute Editor.

  The *paintBlinn* node is added as a second layer of the blue swatch in the layer of the shading group.



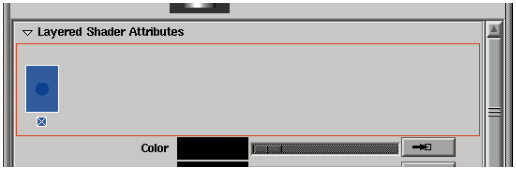
layeredShader: wallLayered

Compositing Flag: Layer Shaders

Material node added to the layered shader

- Click on the boxed "X" below the green swatch.

  This deletes the default layer of the shading group. The *paintBlinn* is now the only layer of the *wallLayered* node.



Default node deleted

Note: If you click in the space of the boxed region for the Layered Shader Attributes section, a new layer is created with the default green. Material or texture nodes can be mapped onto this layer instead of dragging from the Hypershade.

4 Map an image stencil to the color

- Open the *paintBlinn* node's Attribute Editor.

- Click on the **Map** button for the **Color** attribute.
- In the **2D Textures** section of the Create Render Node window:

  Turn **With New Texture Placement** on;

  Select **As stencil**;

  Create a new **File** node.

  A *stencil1* utility node and a *file1* node have been added to the shading group.

5 Reference an image file

- Open the Attribute Editor for the *file1* node.
- Click on the **Browse** button next to **Image Name**.

  The file browser window opens.
- In the *sourceImages* directory, open the *logoStencil.tiff* file.
- In the **Effects** section:

  Set **Filter** to **0.2**.

  The logo image is mapped as a texture onto the material.

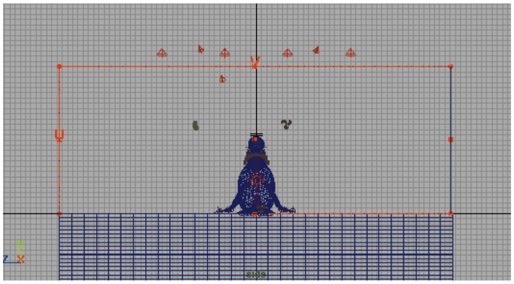
6 View the shading group in the Hypershade

In the Hypershade window, one of the texture placement nodes connects to both the stencil node and to the second texture placement node.

- Press the **Ctrl** key and double-click on the shared texture placement node.
- Enter the new name *placeLogo.*

7 Position the image

- Open a Side view panel then dolly until you can see the entire back wall.
- Select the *placeLogo* node in the Hypershade
- In the Attribute Editor, change the **Rotate UV** to **-90.0**.
- Click the **Interactive Placement** button.



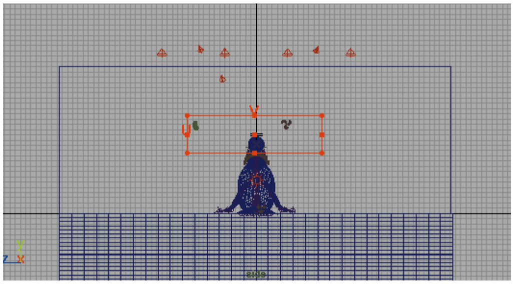
Texture interactive placement manipulator

The wall is outlined with the red interactive placement manipulator.

- With the middle mouse button, click-drag the red square, at the midpoint of one of the surface edges, to a new position.

  The position of the image on the Multilister swatches reflects the manipulation and the values in the Attribute Editor change.
- Adjust the four sides of the manipulator until the texture is positioned as shown.

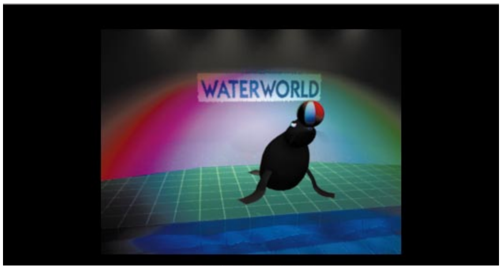
PROJECT FIVE



Stencil texture positioned

Note: If you have hardware texturing, you can barely see the changes on the surface.

- In the Render View panel, select **Redo Previous Render**.



Test render of stencil

8 Create the image mask

In order to remove the white areas of the logo, a grayscale image file can be used as a mask for the stencil node.

By using an image as a mask, you isolate part of the texture to be used and part of the texture to be ignored.

When associated with the stencil, the black areas of the mask represent areas of the image that you want to keep while white areas represent areas that you want to be ignored. Gradations of gray would create a partial masking of the image.

- In the Hypershade, select **Create → Create Render Node...**
- In the Create Render Node Window:

  Turn **With New Texture Placement** off;

  Set the **2D Texture** type to Normal;

  Create a **File** texture node.

  The new *file2* node appears in the Textures section of the Hypershade.
- In the Attribute Editor, click on the **Folder** icon next to **Image Name**.
- **Open** the *logoMask.tiff* image file.
- In the **Effects** section:

  Set **Filter** to **0.2**.

9 Connect the stencil to the shading group

- Open the Attribute Editor for the **stencil1** utility icon.
- Drag the new mask *file2* node onto the **Mask** attribute in the **Stencil Attributes** section.
- In the Hypershade, drag the *placeLogo* node onto the *file2* mask node.

  The mask now shares the same positioning as the logo image and the white area disappears from the material node.
- View the shading network in the Hypershade.

The *placeLogo* node is connected to all of the texture nodes. You only need to adjust the one node to reposition the logo on the surface of the wall.

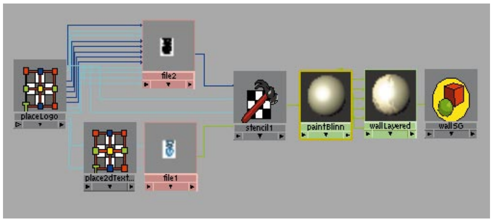


Image of Hypergraph

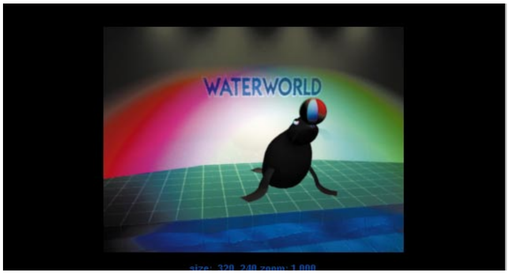
Note: There are more connections between the *placeLogo* and mask file nodes because of the way the connections were made.

10 Modify the paint color

Because the logo has been mapped onto the material node as a stencil, the background color of the material must be adjusted in the stencil node and not on the material node.

- Open the Attribute Editor for the *stencil1* utility node.
- Open the **Color Balance** section.
- Change the **Default Color** swatch to an appropriate color.

  The base color of the shader updates on the Hypershade swatches.
- Re-render the scene to see the effect of the shading group to this point.



Rendering of the stencil logo

Note: If the proportions of the logo do not look correct, you can adjust them by adjusting the placement on the *placeLogo* node.

Adding to the layered shader

The first layer of this shading group represents the painted wall. You will now add a second layer and use a map to set the transparency between them. The lower level will then be textured to look like wood.

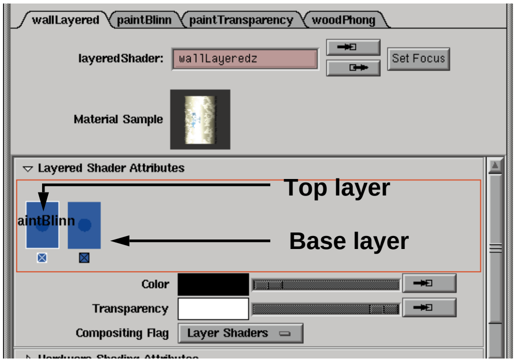
1 Create a base material node

- **Create** a new **Phong** material node without a shading group node.
- Rename the node *woodPhong*.
- Open the Attribute Editor for the *wallLayered* node.
- Drag the *woodPhong* node from the Hypershade to the boxed area of the **Layered Shader Attributes** section.

Note: This is the same result as dragging it onto the *wallLayered* swatch in the Multilister.

- Move the cursor over the two blue swatches to highlight the names.

  The *paintBlinn* swatch to the left is the top layer; the *woodPhong* swatch to the right is the base, or underneath layer of the layered shader. The order of the layers can be changed by dragging the swatches to reposition them.



layeredShader: wallLayeredz

Material Sample

Color

Transparency

Compositing Flag: Layer Shaders

Multiple layers in the Layered Shader Attributes

2 Create a transparency node

This node will be used to create the visual transition between the two layers.

- Create a 3D **Marble** texture node with **New texture placement** still set to **On**.
- In the Attribute Editor:

  Rename the marble node *paintTransparency*;

  Change the **Filler Color** to black;

  Change the **Vein Color** to white;

  Set **Vein Width** to **0.2**;

  Set **Diffusion** to **0.25**;

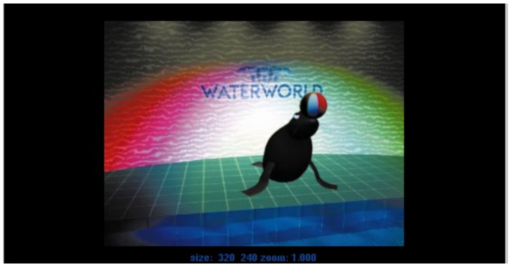
  Set **Contrast** to **0.8**.

- Open the Attribute Editor for the *wallLayered* node.
- Click to highlight the *paintBlinn* swatch in the **Layered Shader Attributes** section.
- Drag the *paintTransparency* swatch from the Hypershade onto the **Transparency** attribute.

  The *wallLayered* material references the marble node – instead of the transparency information within the *paintBlinn* node – to create the transparency between the top and base layers of the shader.

- Re-render the scene to see the effect of the texture.

  Oops! The grain of the marble seems too intense. You need to adjust how this texture is placed.



Default marble texture placement

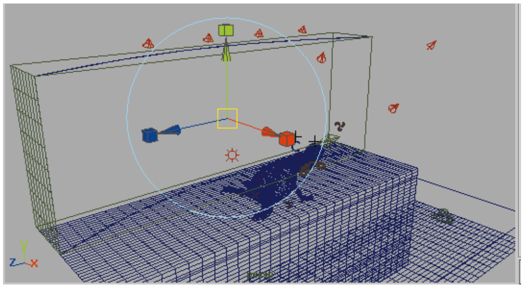
3 Position the 3D texture

Because the marble is a 3D texture, it can be placed using the texture's 3D placement icon. This icon represents how the texture relates to objects in 3D space.

- Open the Attribute Editor for the *paintTransparency* node's 3D texture placement node.

- In the **3D Texture Placement Attributes** section, click on the **Fit to Group box** button.

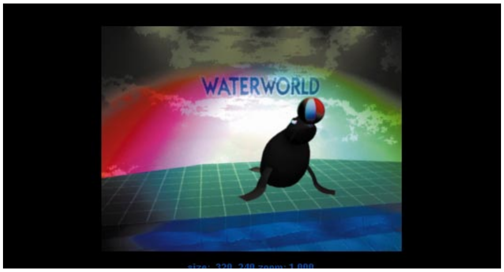
  The 3D texture placement icon moves and scales to surround the surface that the shading group is assigned to. Shown below is the icon as it would appear in the Perspective view panel.



3D placement icon scaled to fit geometry

  Notice that the texture swatch in the Hypershade changes to reflect the change in positioning.

- Re-render the scene to see the effect of the new texture placement.



Transparency texture with new placement

4 Create the base wood texture

- Open the attributes for the *woodPhong* node.
- In the **Specular Shading** section:

  Set **Cosine Power** to **5.0**;

  Set **Specular Color** value to **0.2**;

  Set **Reflectivity** to **0.0**.

- Map a **Grid** texture onto the **Color** attribute.

Note: Make sure that you create it with a new texture placement node.

- Rename the texture node *woodTexture*, and the placement node *woodPlacement*.

5 Set the texture parameters and placement

- Open the attributes for the *woodTexture* node:

  Set the **Line Color** to a pale brown (**36 0.25 0.6**);

  Set **U Width** to **0.0**;

  Set **V Width** to **0.025**.

  The **Filler Color** will be mapped with another texture so it is not necessary to set its color.

- Open the attributes for the *woodPlacement* node.

  Set the **Repeat V** to **30.0**.

  The **Repeat U** can remain unchanged because the lines in this direction will not be visible.

6 Map a brownian texture

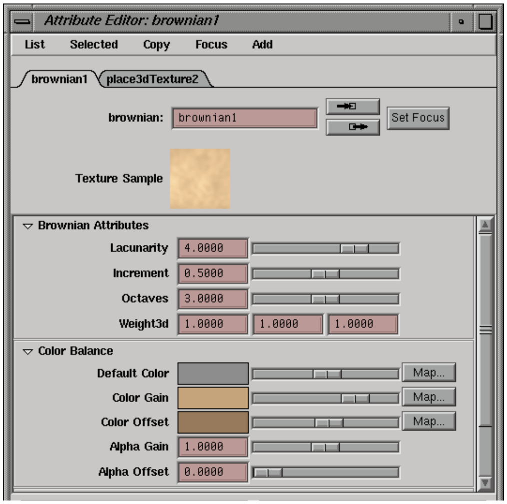
The brownian texture is used to create a streaked wood texture. Because the wall is in the background and covers a large area, an abstracted texture is more suitable than the procedural wood texture.

- Open the attributes for the *woodTexture* node.
- Click on the **Map** button for the **Filler Color** attribute.
- In the **3D Textures** section select a **Brownian** texture.
- In the **Brownian Attributes** section of the Attribute Editor:

  Set **Increment** to **0.5**.
- In the **Color Balance** section:

  Set the **Color Gain** to a light brown (**30 0.4 0.75**);

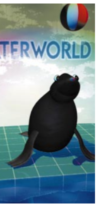
  Set the **Color Offset** to a slightly darker brown (**30 0.4 0.53**).



brownian: brownian1

Lacunarity	4.0000		
Increment	0.5000		
Octaves	3.0000		
Weight3d	1.0000	1.0000	1.0000
Alpha Gain	1.0000		
Alpha Offset	0.0000		

Brownian texture



7 Position 3D texture placement

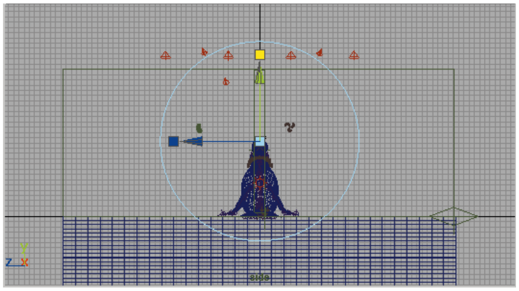
The fill color texture can be transformed into a streaked directional texture by adjusting the texture placement node.

Because the brownian node is a 3D texture, the texture placement can be manipulated as an object in the 3D XYZ space of the scene, instead of on the 2D UV space of the surface.

- In the Hypershade, select the brownian node's *place3Dtexture* node.
- In the 3D Texture Placement Attributes section, click on the **Interactive Placement** button.

  A combined move, scale and rotate manipulator is displayed. The icon by default is a single unit cube located at the origin.
- **Scale** the icon along the Y-axis.

  It should be about the same height as the back wall.
- **Move** the icon so that it is at the approximate center of the back wall.

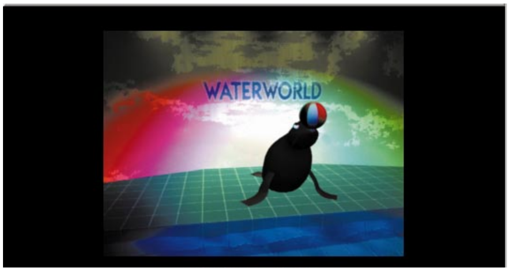


3D placement node scaled and moved

Note: Because the 3D texture in this case is fairly consistent, the specific position of the icon is not as important as the fact that it is scaled or stretched. In other cases, the specific position, scale and rotation of the 3D placement icon is very important.

8 Render scene

- Re-render the scene.



Rendering of finalized materials

REFINING THE LIGHTING

To refine the lighting in the scene, you will add a new light that represents light reflecting off of the deck and then you will add shadows to some of the other lights.

Adding reflected lighting

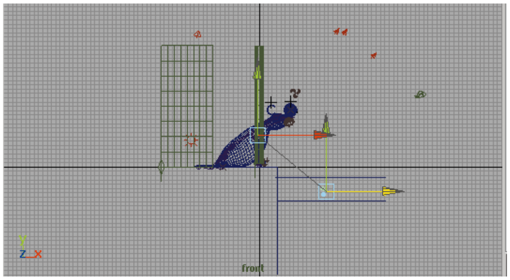
The light in a computer generated scene cannot reflect off surfaces. You can see the reflection of images but light serves only to illuminate the scene.

To simulate the reflection of light off of the water, you will use a new light, then map the animated water texture to the light's color.

1 Add a new spot light

- Select **Lights → Create Spot Light**.
- Position the light so that it is looking up at the seal from just below the surface of the water.

  Use *look through selected* to verify the position. The spot radius should cover the seal and the ball at its highest point.



Spot light view looking up at Salty

2 Add water map as filter to light

- Open the attributes for the new spot light.
- Drag the *waterColor* texture swatch from the Hypershade onto the **Color** attribute.

  The animated texture node is now connected to the spot light and will be projected up onto Salty.
- Set the **Intensity** to **0.5**.

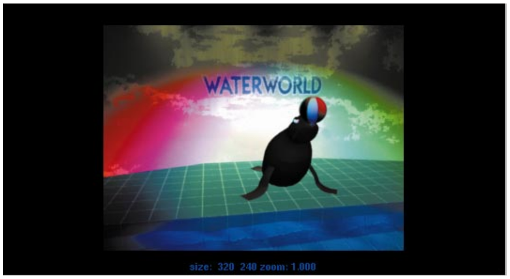
3 Link light to seal and ball

Since this light is not needed on all of the surfaces, you are going to link it to the shading groups belonging to the ball and Salty. This will later make your rendering more efficient because this light is only rendered in relation to the linked shading groups.

- In the Attribute Editor, click on the **Illuminates by Default** button to turn it **off**.

  The light in no longer linked to any of the shaders or objects, and must be specifically linked to the shaders that you want the light to shine on.
- Select all of Salty's surfaces and the ball's surface.

- Select the new spotlight.
- Select **Lights → Make Light Links**.



Lights linked to shaders

Note: In Maya, lights are linked not to the objects but to their shading groups.

Turning on shadows

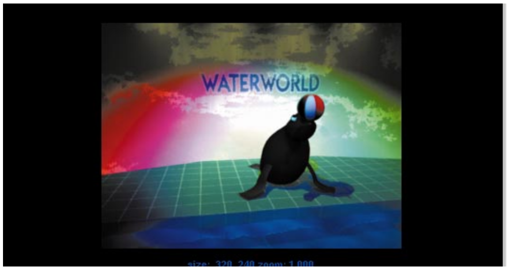
To achieve realistic lighting, shadows become very important. At the same time, they take longer to render and, therefore, must be set with caution. By default, lights do not cast shadows. You can then turn shadows on for lights that are key to the scene.

For Salty, you will use the three main lights for casting shadows while the other lights are used only for illumination.

1 Turn shadows on for the lights

You will use depth map shadows for the three main lights. This shadow type is fast and offers a soft look.

- Select the three colored spot lights.
- In the Channel box, set the **Depth Map Shadows** to **On**.
- Test render the scene.



Rendering with depth map shadows

You will notice that the lights are casting colored shadows. This is a natural phenomenon that is mimicked by colored lights in Maya.

You may want to leave these different shadows or edit the way the scene is lit to achieve a different result. If you increase the value setting for the three lights, then their color will be less prominent and the shadows will be darker.

You could also choose to only cast shadows from the light that produces the best shadow. In this case, the green light is producing the strongest shadow. Another option would be to change the color of the tiled deck which also affects the look of the shadow.

These are the kinds of decisions you need to make during the test rendering of a scene. You'll find that you will have to evaluate the look of the rendering, and tweak and adjust your settings until you are happy with the results.

2 Save your work

RENDERING

You are now going to batch render the scene. Rendering an animation offers you the chance to see the fruits of your labor. Before proceeding, you should prepare your surfaces and set the Render Globals.

Preparing the surfaces

To render as quickly as possible with the best quality possible, you should look at how to subdivide your polygonal and NURBS surfaces.

1 Reduce the polygon subdivisions

- **Select** the deck surface.
- Click on the surface's shape node, then the *polyPlane* input node, and set the following:

  Subdivisions X to **1**;

  Subdivisions Y to **1**.

  The higher settings made it possible to preview the lighting on the surface; however, rendering only one subdivision in each direction is better because less geometry will render faster.
- Repeat the same edit for the input nodes of the other polygonal surfaces.

2 Set the NURBS tessellation

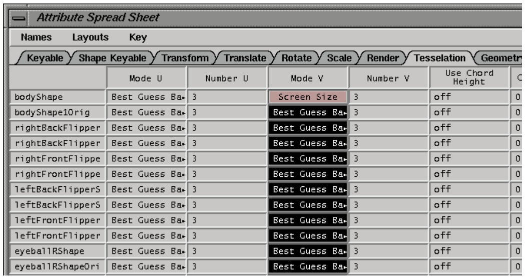
Earlier, you set the tessellation of the water surface to make sure that there was enough detail when you rendered the displacement. You should now set the tessellation for other NURBS surfaces.

Tessellation is the method by which your NURBS surfaces are broken down, or subdivided, into triangles for rendering. If you use a higher tessellation, the results are better but the rendering is slower. Maya offers you a *Best Guess* option that lets the renderer choose the appropriate tessellation.

- **Select** the surfaces that make up Salty, the ball surface and the back wall.
- Select **Window** → **General Editors** → **Attribute Spread Sheet...**
- Click on the tabs at the far right and choose **Tessellation** from the pop-up.
- Under the **Mode U** column, click-drag over all the entries.

  By selecting all of this column, you can edit one value to update them all.
- Type **4** to enter the mode option then press the **Enter** key.

  This value sets the tessellation to **Best Guess Based on Screen Size**. This means that the renderer will tessellate your scene based on the size that the object fills the screen. This offers you an optimized tessellation solution.
- Repeat for the *Mode V* column.



	Mode U	Number U	Mode V	Number V	Use Chord Height	C
bodyShape	Best Guess Ba	3	Screen Size	3	off	0
bodyShape1Orig	Best Guess Ba	3	Best Guess Ba	3	off	0
rightBackFlipper	Best Guess Ba	3	Best Guess Ba	3	off	0
rightBackFlipper	Best Guess Ba	3	Best Guess Ba	3	off	0
rightFrontFlippe	Best Guess Ba	3	Best Guess Ba	3	off	0
rightFrontFlippe	Best Guess Ba	3	Best Guess Ba	3	off	0
leftBackFlipperS	Best Guess Ba	3	Best Guess Ba	3	off	0
leftBackFlipperS	Best Guess Ba	3	Best Guess Ba	3	off	0
leftFrontFlipper	Best Guess Ba	3	Best Guess Ba	3	off	0
leftFrontFlipper	Best Guess Ba	3	Best Guess Ba	3	off	0
eyeballRShape	Best Guess Ba	3	Best Guess Ba	3	off	0
eyeballRShapeOri	Best Guess Ba	3	Best Guess Ba	3	off	0

Attribute Spread Sheet window

Note: The number 4 represents the fourth option if you were to look up Mode U in the tessellation control section of the Attribute Editor.

Setting the Render Globals

You have already used the Render Globals window to set up some test renderings. This window is where you make your final decisions before sending a rendered animation off to the batch renderer.

1 Open the Render Globals window

- Select **Window** → **Render Globals...**

2 Set the Camera's render status

- In the **Image File Output** section set **Camera** to **camera1.**
- Set **Channels** to **RGB Channel (Color)** to On.

  These settings mean that this camera will be considered by the renderer. If you were compositing, you might also turn on the mask layer but, for now, only the image is required.

3 Turn on animation and motion blur

By default, the Render Globals are set up to render the current frame. If you want to render an animation, then you have to turn animation on.

- Set **Frame/Animation Ext** to the following:

  name.ext.# for **IRIX;**

  name.#.ext for **NT.**

- Set the following:

  Start frame to **1**;

  End frame to **450**.

4 Set the quality

- Click on the Anti-alias Quality section.
- Set **Presets** to **Production quality**.

5 Set the size

- In the Resolution Section, set **Resolution** to **320x240.**

6 Turn on motion blur

- In the Motion Blur section, click turn **Motion Blur** to **On**.
- Set **Motion Blur Type** to **2D**.

7 Save your work

- Select **File** → **Save Scene As**... Enter the name *Salty6* and press **Save**.

8 Batch render the scene

- Select **Render** → **(Save) Batch Render**.
- Enter the name *SaltyRender* and click on the **Batch Render** button.

Look at the Feedback line in the lower right to follow the rendering progress.

Tip: At this point, you could exit Maya and the rendering would continue. This is because the renderer is a separate application. If you do this, then you will lose the feedback on the renderings progress but you will free up RAM for the renderer. You can then use *fcheck* to check the progress of each frame.

9 Preview a frame

- Select **Render** → **Show Batch Render**.

This shows you the progress on the frame being rendered right now.

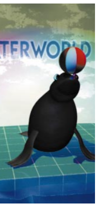
10 Preview the animation

Once you have completed the animation, you can use Playblast or *fcheck* to view the movie.

Conclusion

Congratulations, you now have a fully animated and articulated character. You have learned how to work with a wide range of Maya's feature set in an integrated project.

In the last lesson, you will explore Maya's scripting language, MEL as you set up custom controls for blinking Salty's eyes.


TERWORLD

23 Blinking using MEL

In this lesson, you will set keys on the blink attribute that you created on the *eyeControl* node in Lesson 19. To help with this task, you will create a MEL (Maya Embedded Language) script that will help you animate the blink.

MEL is a powerful scripting language that can be used by both technical directors and animators to add to what Maya can do. Animators will be able to take advantage of simple *macro-like* scripts to enhance their workflows, while technical directors can use more advanced MEL commands to rig up characters, add special visual effects, or set up customized controls.

If you know nothing about programming and scripts, this lesson will, at first, seem foreign to your world of graphics and animation. While you can certainly be successful with Maya without relying on the use of MEL, this lesson offers a good chance to get your feet wet and see the possibilities. If you do learn how to use MEL, you will be quite surprised how a simple script can be used to enhance your work.

In this lesson you will learn the following:

- How to recognize and enter MEL commands
- How to create a MEL script procedure
- How to use this procedure within Maya's existing UI
- How to build a custom UI element for the procedure
- How to animate Salty's blinking using the procedure

Starting a new file

Rather than working in the Salty scene file, you will practice using MEL in a new file. Once your scripts have been written and saved, you will return to the Salty scene and use the custom user interface tools in context.

1 Start a new file

- Select **File** → **New Scene**.
- Set up a single Perspective view panel.
- Make sure that the Command line, the Help line and the Channel box are all visible.

WHAT IS MEL?

MEL stands for Maya Embedded Language. MEL is built on top of Maya's base architecture and is used to execute commands that you use to build scenes and create user interface elements. In fact, every time you click on a tool, you are executing one or more MEL commands.

Typing commands

A MEL command is a text string that tells Maya to complete a particular action. As a user, it is possible to skip the graphical user interface and use these commands directly. Generally, animators will choose the user interface instead – but it is still a good idea to know what MEL can do at a command level.

The Command line

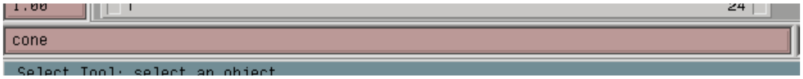
You will now use Maya's Command line to create and edit some primitive objects. The goal at this point is to explore how simple commands work.

1 Create a cone using the Command line

- Click in the Command line or press the ` key to set your *focus*.

  The Command line can be found at the bottom left, just above the Help line.

- Enter the following:



```
cone
```

Select Tool: select an object

Entering a MEL command

- After you finish, press the **Enter** key on the numeric keypad section of your keyboard.

Tip: The keyboard has two **Enter** keys that each work a little differently with the Command line. The **Enter** key associated with the numeric keypad will keep your focus on the Command line, while the **Enter** key associated with the alpha-numeric keyboard switches your focus back to the view panels.

2 Rotate and move the cone with commands

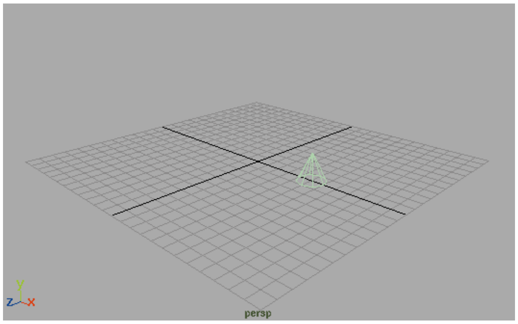
The next step is to transform the cone using MEL commands.

- Enter the following:

`rotate 0 0 90` **< Enter >**

`move 5 0 0` **< Enter >**

You now have a cone sitting on the ground surface, five units along the X-axis. You entered the commands by first entering the command then adding the desired values.



Perspective view of cone

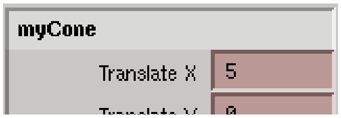
3 Rename the cone

You can also rename objects from the command line.

- Enter the following:

`rename “nurbsCone1” myCone` **< Enter >**

You can look in the Channel box to confirm that the object has been renamed.



myCone

Translate X	5
Translate Y	0

Channel box with cone's name

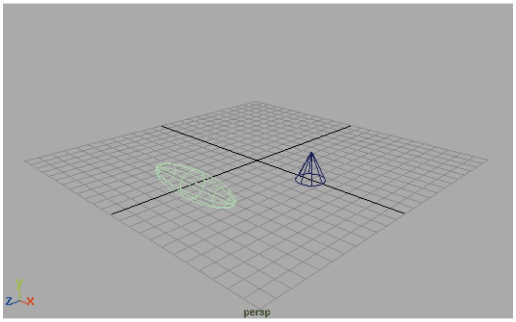
4 Execute two commands at once

If you quickly want to enter more than one command without pressing the enter key along the way, then you can place a semicolon between the commands.

- Enter the following:

```
sphere; move 0 0 6; scale 4 1 1 < Enter >
```

Using the semi-colon(;), you executed three commands in a row. First, you created a sphere, then you moved it, then you scaled it. The semi-colon will become important later when you write scripts.



Perspective view of new sphere

5 Execute a command on an unselected object

If you want to execute a command on an object that is not selected, then you simply have to add the name of the node that you want to affect. The node will follow the command without requiring the cone to be selected.

- Enter the following:

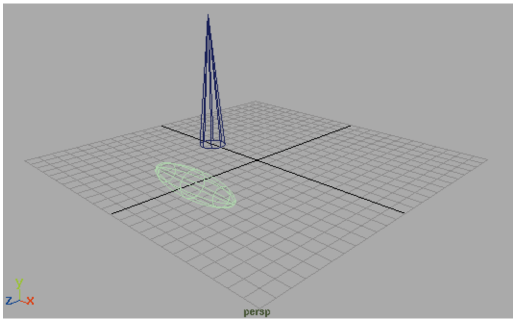
```
move -5 0 0 mycone < Enter >
```

Oops! You get an error message saying that Maya cannot find the `mycone` object. This is because the object name has a capital C for the word 'Cone'. MEL is case sensitive which means you should be especially aware of how you spell and capitalize any names or commands.

- Enter the following:

```
move -5 0 0 myCone < Enter >
scale 5 1 1 myCone < Enter >
```

Always remember the importance of getting the spelling of commands correct. Later, when you write scripts, this spelling will be very important.



Perspective view of edited cone

6 Use commands flags

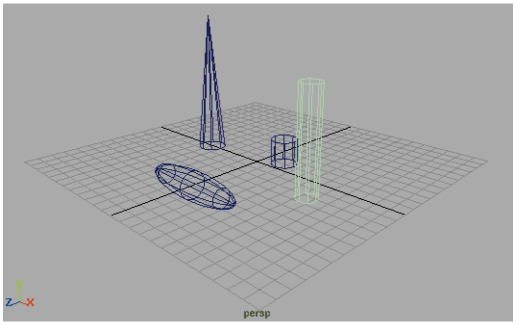
Another important MEL capability is the command flag. You can use these flags to be more specific about how you want the commands to be executed. The command flags can have short or long names. Flags are indicated with a hyphen in your script. Shown below are examples of both kinds of flags.

- Enter the following using long names for flags:

```
cylinder -name bar -axis 0 1 0 -pivot 0 0 -3 < Enter >
```

- Enter the following using short names for flags:

```
cylinder -n bar2 -ax 0 1 0 -p 0 0 -6 -hr 10 < Enter >
```



Perspective view of cylinders

The short flag names represent the following:

`-n`	name
`-ax`	axis
`-p`	pivot
`-hr`	height ratio

Tip: You will notice that long flag names can create a commands that is easy to read but hard to type in – the short names are harder to decipher but easy to type. Generally, entering in the command line is a good place for short flags while long flags should be used in scripts to aid in readability.

7 Delete all objects

- Enter the following:

```
select -all; delete < Enter >
```

The Script Editor window

You may have noticed that the Command line is a small space to work in and only one line of feedback. The Script Editor is a special user interface element that will make entering commands easier.

Up until now, you have been entering random commands in order to learn a little about their syntax and how they work. You will now use the script editor to build a sphere and a locator which will mimic the *eyeLid/eyeControl* relationship that you set in Lesson 19. The ultimate goal is to set up a blink attribute that will control the blinking of the eyelid.

1 Open the Script Editor window

- Click on the Script Editor button in the lower right of the workspace or select **Window** → **General Editors** → **Script Editor**.

  The window opens to show all of the commands you just entered.

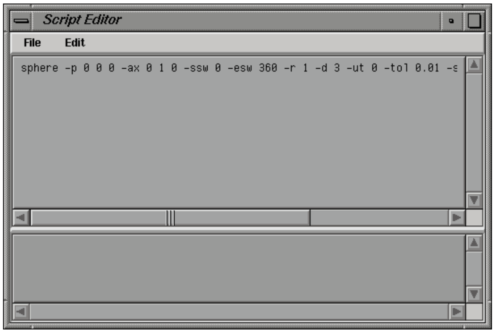
- From the Script Editor, select **Edit** → **Clear History**.

2 Create a primitive sphere using the Primitives menu

In the earlier lessons, you have been using MEL without having to type a single command. Most Maya tools and windows have been designed and implemented using MEL.

- Go to the **Modeling** menu set.
- Select **Create** → **NURBS Primitives**→ **Sphere**.

  In the Script Editor, you can see the MEL command that was used to create the sphere. Also included are the flags with default settings presented in their short form.



```
sphere -p 0 0 0 -ax 0 1 0 -ssw 0 -esw 360 -r 1 -d 3 -ut 0 -tol 0.01 -s
```

Script Editor

- In the lower portion of the Script Editor, type the following:

```
delete
```

- Press the numeric keypad **Enter** key to execute the command.

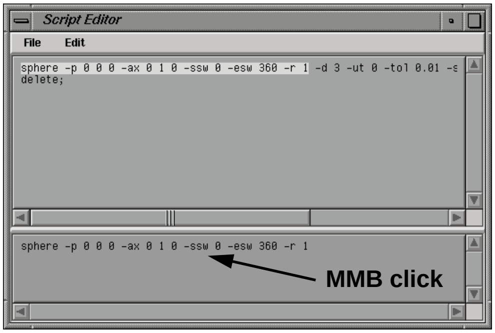
Tip: In the Script Editor, the numeric keypad’s **Enter** key executes an action while the alpha-numeric keypad’s **Enter** key returns to the next line.

---

3 Copy and edit the sphere commands

Now that the sphere command is in the Script Editor's history, you can use this command as a start point to write your own command.

- In the Script Editor, select the parts of the Command line up to the `-r 1` flag.
- Move your cursor to the edit area below and click with your middle mouse button. This places the selected text in this part of the editor.



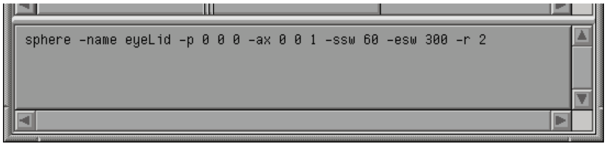
```
sphere -p 0 0 0 -ax 0 1 0 -ssw 0 -esw 360 -r 1 -d 3 -ut 0 -tol 0.01 -s
delete;
```

```
sphere -p 0 0 0 -ax 0 1 0 -ssw 0 -esw 360 -r 1
```

Script Editor

- Edit the first part of the command to read as follows:

```
sphere -name eyeLid -p 0 0 0 -ax 0 0 1 -ssw 60 -esw 300 -r
2
```



```
sphere -name eyeLid -p 0 0 0 -ax 0 0 1 -ssw 60 -esw 300 -r 2
```

Edited commands

- Press the **Enter** key on your numeric keyboard.

4 Create a locator

- Select **Primitives** → **Create Locator**.

  In the Script Editor, you do not see a corresponding MEL command.

- Enter undo to go back one step.

5 Echo all commands

- In the Script Editor, select **Edit** → **Echo All Commands**.
- Select **Create** → **Locator**.

  In the Script Editor, you can now see a MEL command that you can use to create a locator – `createPrimitive nullObject;`

Note: This command is surrounded by other commands that belong to Maya. You only need to focus on the locator command.

- In the Script Editor, select **Edit** → **Echo All Commands** to turn this option off.

6 Rename and move the locator

You will now name the locator as *eyeControl*. This object will be used as a substitute for the control node you built earlier in the Salty scene.

- Enter the following:

```
rename “locator1” eyeControl;
move 10 0 0 eyeControl  < Enter >
```

7 Add an attribute to the locator

You will now add a blink attribute. This command is the same as using **Modify** → **Add Attribute** from the UI.

- Enter the following:

```
addAttr -ln blink -at “float”
-min 0 -max 10 -dv 10 eyeControl < Enter >
```

The short flag names represent the following;

`-ln` long name of the new attribute

`-at` attribute type

`-min/max` minimum/maximum values for the attribute

`-dv` default value for the attribute

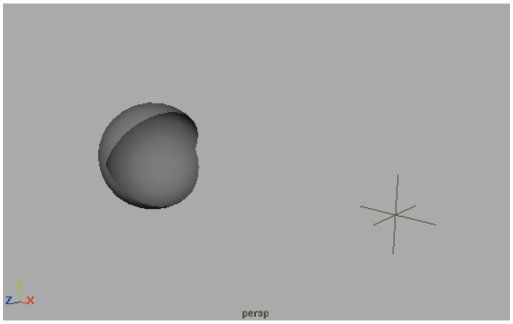
8 Make the attribute keyable on the blink attribute

- Enter the following:

```
setAttr -keyable true eyeControl.blink < Enter >
```

9 Set up your Perspective view panel

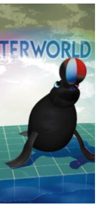
- Set up your view panel so that you can see the eyelid object and the locator.
- Select the eyelid and press **3** to increase the smoothness and **5** to turn on hardware shading.
- Select **Display** → **Grid** to turn off the grid.



Eyelid and locator

Learning more about commands

You now know how to use a few of the many Maya commands. To learn more about the commands, refer to the online documentation where you will find a complete list of all the commands available in MEL. Each command is listed with descriptions of the various flags.



Expressions

When you write an expression in the Expression Editor, it can be written as a MEL script. You can also use MEL to create the expressions from within the Script Editor.

You will create two expressions to control the *start* and *endSweep* of the eyeLid's *makeNurbSphere* input node. In Lesson 19, you used Set Driven Key. Now you can compare the use of expressions to that earlier approach.

1 Add an expression to the eyelid’s startSweep

This expression is designed to make sure that as the blink attribute goes from 0 to 10, the eyelid rotates 60 degrees. Therefore, the blink attribute is multiplied by 6.

- Enter the following:

```
expression -n toplid -o makeNurbSphere1
-s “ssw = eyeControl.blink * 6” < Enter >
```

2 Add an expression to the eyelid’s endSweep

This expression is similar to the last one except now you want the eyelid to rotate from 360 to 300 degrees. Therefore the blink attribute is multiplied by 6 then subtracted from 360.

- Enter the following:

```
expression -n bottomlid -o makeNurbSphere1
-s “esw = (360 - eyeControl.blink * 6)” < Enter >
```

3 Test the blink attribute

- Enter the following:

```
setAttr “eyeControl.blink” 5 < Enter >
```

4 Set keys on the blink attribute

- Enter the following:

```
setKeyframe -at blink -t 1 -v 10 eyeControl;
setKeyframe -at blink -t 5 -v 0 eyeControl;
setKeyframe -at blink -t 10 -v 10 eyeControl < Enter >
```

The short flag names represent the following:

`-at` attribute that is being keyed
`-t` time at which you want the key set
`-v` value of the attribute that you want to key

- Play back the results.

Keys have been set at several frames with the eye opening and closing to match.

Building a blink procedure

You are now going to create a blink procedure that you will save as a MEL script. The next few steps outline all of the parts of the MEL script with some tips for how to enter it and how to execute it. At the end of this lesson, you will find the script without descriptive text. You can enter the script later, in case you want to read over this section first.

Writing the script

You will write the blink procedure not in the Script Editor but in a text editor. A text editor is an application that lets you quickly work with text then save in a generic text format.

1 Open a text editor

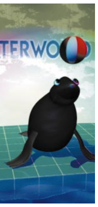
- Go from the **Start** menu to your Windows NT accessories.
- Open a text editor such as *Notepad or WordPad*.

2 Type comments to create header information

Every script should start with a header that contains important information for other people who might read your script later. Below is an example header. The // that is placed in front of these lines indicate that these lines are comments and therefore will be ignored when you later execute the script.

- Type the following:

```
//
// Creation Date:      Today’s date
// Author:             Your name here
//
// Description:
//       Learning Maya tutorial script
//       This script builds a procedure for animating
//       Salty the seal’s eyeControl.blink attribute
//
```



Tip: Please don’t underestimate the importance of commenting your scripts. Down the line, someone will need to read what you have done and the comments are much easier to follow than the actual script.

3 Declare the procedure

The first thing you will enter is designed to declare the procedure. This line loads the procedure into Maya’s memory so that it can be executed later.

- Type the following:

```
global proc blink (float $blinkDelay){
```

This line defines a procedure named *blink*. The required argument is within the round brackets. This tells Maya what the script requires to execute. In this case, the length of the blink action is required. This is defined as a floating value called `$blinkDelay`. Because this value is not yet determined, it is known as a variable. The $ sign defines it as a variable. The open bracket – the { symbol – is added to the end of the declaration to let you start inputting MEL statements.

4 Set up variables

Within your script, you will use variables to represent values that may need to change later. At the beginning of the script, you need to set up the variables and set their value. In some cases, you may set their value with an actual number. But for this script, you will use attribute names and values instead.

- Type the following:

```
// Set up variables that will be used in the script

   string $blink = "eyeControl.blink";
   float $time = `currentTime -query`;
   float $blinkCurrent = `getAttr $blink`;
```

The first variable set defines *$blink* as the blink attribute found on the *eyeControl* node. The second variable queries Maya for the current time.

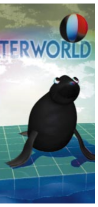
The third attribute gets the actual value of the *eyeControl.blink* at the queried time.

5 Set keys on the blinking

Next, you want to set keys on the blink attribute at the beginning, middle and end of the blink. The length of the blink will be defined by the *blinkDelay* variable that was set out as the main argument of the procedure. Notice that, while other variables were set at the beginning of the script, the *blinkDelay* is used as an argument so that you can set it when the script is executed later. As you enter the keyframe commands, notice how you use the normal setup of command/flag/node name.

- Type the following:

```
// set key for the blink attribute
// at the current time

   setKeyframe -value $blinkCurrent
         -time $time
         -attribute $blink
         $blink;

// set key for a blink of 0
// half way through the blink

   setKeyframe -value 0
         -time ($time + $blinkDelay/2)
         -attribute $blink
         $blink;

// set key for the original blink value
// at the end of the blink

   setKeyframe -value $blinkCurrent
         -time ($time + $blinkDelay)
         -attribute $blink
```



```
            $blink;

      }
```

In this part of the script, you have set keys using the *setKeyframe* command. The keys set at the beginning and end of the blink use the queried value of the blink attribute, while the key set in the middle uses a value of zero. At the end, a closed bracket – the } symbol – is used to declare the statement complete.

6 Save your script

You can now save your script into your Maya scripts directory. This will make sure that the procedure is easily available within Maya any time you need it.

- In your text editor, save the script using the following path:

  ```
  \[drive]:\maya\scripts\blink.mel
  ```

Note: Because the procedure is named *blink*, it is important to save the file as *blink.mel*. This helps make it easier for Maya to find the function.

7 Testing the script

Because you named the file *blink.mel* and placed it in your *maya/scripts* directory, you can execute the command in Maya without having to load the script. If you enter `blink` with a value for the blink delay, then Maya will look in the scripts directory for a procedure called *blink.mel*.

- Set the **Time slider** to frame **40**.
- Enter the following:

  `blink 10` **< Enter >**

- Scrub in the Time slider to test the results.

  If this works, then you can congratulate yourself on completing your first MEL script and you can move on to the next section.

  If it doesn't, then you typed something incorrectly and MEL is finding your mistake.

8 Debugging your script

To debug your script, you need to find out which line is causing the error, and then you need to go back and check your spelling and syntax. Did you use the correct symbols? Did you name your nodes correctly? Is your capitalization correct? These and similar questions are what you need to ask yourself.

- To display line numbers in the Script Editor, select **Edit → Show Line Numbers.**

Adding the function to the UI

Now that you have created your own function, you will want to have easy access to it. Below are three methods for adding your function to the default UI, which you can easily set up using interactive methods.

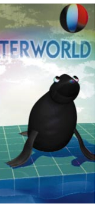
1 Creating a shelf button

- In the Script Editor, select the text `blink 10`.
- Click on the selected text with the middle mouse button and drag up to the shelf.

  It is placed on the shelf with a MEL icon. You can now move the Time slider to a new position and test it. You could also drag up different *blinkDelay* settings to offer different blink settings. Or you could set up a marking menu as outlined below.

2 Creating a blink marking menu set

- Select **Options → Customize UI → Marking Menus...**
- Click on the **Create Marking Menu** button.
- Click on the top middle square with your right mouse button and select **Edit Menu item...** from the pop-up menu.
- In the Edit window, type *Blink 10* in the **Label** field.
- In the **Commands(s)**: field, type `blink 10`**.**
- Click **Save** and **Close**.
- Repeat for the other quadrants to set up blink commands that use a *blinkDelay* of 20, 30 and 40.
- In the **Menu Name** field, enter: `blinking`.



3 Prepare the blink marking menu for a hotkey

The blink marking menu now needs to be set up.

- In the marking menu customize window, set the following:

  Use Marking Menu in to **Hotkey Editor.**

  Now the marking menu can be set up in the Hotkey Editor so that it can be evoked using a hotkey.

- Click the **Apply Settings** button then **Close**.

4 Assign the blink marking menu to a hotkey

- Select **Options** → **Customize UI** → **Hotkeys...** and set:

  Display Mask to **Show All**.

- Scroll to the bottom of the list and click on the **blinking (Press)** listing.
- In the **Key Settings** section, set the following:

  Key to **b**;

  Action to **Press**.

- Click on the **Query Key** button to verify that this key is not in use.
- Press the **Apply New Settings** button.
- Scroll to the bottom of the list and click on **blinking (Release)**.
- In the **Key Settings** section, set the following:

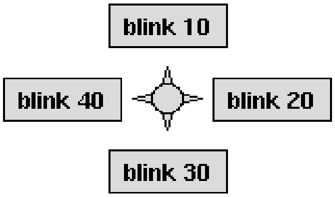
  Key to **b;**

  Action to **Release.**

- Press the **Apply New Settings** button then Close.

5 Use the new marking menu

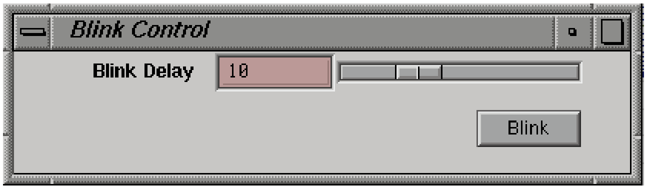
- Go to frame **80**.
- Click and hold on the **b** key, then pick one of the blinking options from the marking menu.



Blink marking menu

Building a custom UI script

In the next section, you will write a second script that will build a custom user interface window that includes a slider for the *blinkDelay* variable and a button that executes the blink procedure you scripted earlier. In Maya, you have the ability to use MEL to build custom user interface elements.



Custom user interface window

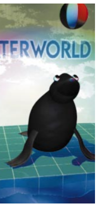
1 Start a new text file

2 Adding the opening comments

Start the script with a commented header that helps others read your work. While this was taught earlier, it can never be emphasized enough.

- Type the following:

```
//
// Creation Date::     Today's date
// Author:             Your name here
//
// Description:
//       Learning Maya tutorial script
//       This script builds a custom user interface
//       for executing the blink procedure
//       and for setting the blink delay
//
```



3 Declare a get info procedure

You are now going to create a procedure called *blinkGetInfo* that will be used to get the *blinkDelay* value from a slider, which you will build later in the script. Since the value set in the slider is meant to be the chosen value for the blink, this procedure queries the slider to set the *blinkDelay*, and then adds that value next to the blink command.

- Type the following:

```
global proc blinkGetInfo() {

   // get necessary information from Maya

   float $blinkDelay = ‘intSliderGrp
   -query -value blinkWindow|columnLayout|delaySlider‘;

   blink $blinkDelay;

}
```

4 Declare a second user interface procedure

You are now going to declare a procedure that will build a floating window. This window will look and act like any other window in Maya but will be designed to help you put a blink to Salty’s eyes.

- Type the following:

```
global proc blinkWindow() {
```

5 Remove any existing blink windows

As you start a user interface script, it is a good idea to check if the same UI element already exists in the scene and, if so, then to delete it. This makes sure that your new element is the only one recognized by Maya at any one time.

- Type the following:

```
// clean up any existing blinkWindows
if ( (`window -ex blinkWindow`) == true ) deleteUI
blinkWindow;
```

6 Build the window called blinkWindow

The next part of the script is designed to build a window that is 400 pixels wide and 75 pixels tall. You will call it Blink Control in its title bar but Maya will know of it as *blinkWindow*.

- Type the following:

```
window
   -width 400
   -height 75
   -title "Blink Control"
blinkWindow;
```

7 Form a column layout

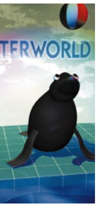
Within the window, you need to organize your user interface elements. One method of organization is a *columnLayout*. This sets up a column with a particular spacing in relation to the window.

- Type the following:

```
columnLayout
   -columnAttach "right" 5
   -rowSpacing 10
   -columnWidth 375
columnLayout;
```

8 Create a slider group

Within the layout, you want to build a slider that lets you set the *blinkDelay* value. MEL offers you preset *kits* using special group commands that build several UI types in one go. The *intSliderGrp* builds a slider along with a field for seeing the resulting value and for entering the value yourself. This slider is set to integer values since frames are generally set in whole numbers. The flags let you set the various values for the minimum and maximum settings of the slider.



- Type the following:

```
intSliderGrp
   -label "Blink Delay"
   -field true
   -minValue 2
```

```
      -maxValue 30
      -fieldMinValue 0
      -fieldMaxValue 100
      -value 10
   delaySlider;
```

9 Create a button

The next part of the script builds a button that you will be used to execute the *blinkGetInfo* procedure, which in turn uses the *blinkDelay* value from the slider to execute the *blink* command. At the end, you will enter *setparent* to link the button to the *columnlayout*.

- Type the following:

```
button
   -label "Blink"
   -width 70
   -command "blinkGetInfo"
button;

   setParent ..;
```

10 Show the window

You are almost finished, except that you must tell Maya to show the window.

- Type the following:

```
showWindow blinkWindow;
```

11 Finish the script

Finally, you must complete the procedure then make one final declaration of the *blinkWindow* procedure name.

- Type the following:

```
}

blinkWindow;
```

12 Saving the script

You can now save your script into your Maya scripts directory.

- In your text editor, save the script using the following path:

```
\[drive]:\maya\scripts\blinkWindow.mel
```

13 Test your script

- In the Command line or the Script Editor, type the following:

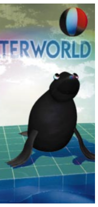
`blinkWindow` **< Enter >**

The window should open. You can now set the Time slider to a new time, and then set the blink delay using the slider; pressing the button will key the blink.

Keyframing Salty's blink

Congratulations! You now have your own custom user interface element built and ready to go. You can now open your Salty file from the last lesson and use this script to make her blink. Note that this will only work if you named your `eyeControl` node correctly and created a `blink` attribute as taught.

THE SCRIPTS



Here are the two scripts listed in their entirety for you to review.

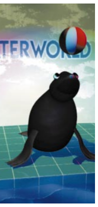
blink.mel

```
//
// Creation Date:      Today's date
// Author:             Your name here
//
// Description:
//       Learning Maya tutorial script
//       This script builds a procedure for animating
//      Salty the seal’s eyeControl.blink attribute
//

global proc blink (float $blinkDelay){

// Set up variables that will be used in the script

   string $blink = "eyeControl.blink";
   float $time = `currentTime -query`;
   float $blinkCurrent = `getAttr $blink`;

// set keys for the blink attribute

   setKeyframe -value $blinkCurrent
         -time $time
         -attribute $blink
         $blink;

   setKeyframe -value 0
         -time ($time + $blinkDelay/2)
         -attribute $blink
         $blink;

   setKeyframe -value $blinkCurrent
         -time ($time + $blinkDelay)
         -attribute $blink
         $blink;

   }
```

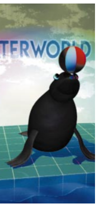
blinkWindow.mel

```
//
//Creation Date:Today’s date
//Author:Your name here
```

```
//
//Description:Learning Maya tutorial script
//      This script builds a custom user interface
//      for executing the blink procedure
//      and for setting the blink delay
//

global proc blinkGetInfo() {

        // get necessary information from Maya

        float $blinkDelay = `intSliderGrp
   -query -value blinkWindow|columnLayout|delaySlider`;

   blink $blinkDelay;

}

global proc blinkWindow() {
   if ( (`window -ex blinkWindow`) == true ) deleteUI
blinkWindow;

   window
      -width 400
      -height 75
      -title "Blink Control"
   blinkWindow;

   columnLayout
      -columnAttach "right" 5
      -rowSpacing 10
      -columnWidth 375
   columnLayout;
```



```
   intSliderGrp
      -label "Blink Delay"
      -field true
        -minValue 2
      -maxValue 30
      -fieldMinValue 0 -fieldMaxValue 100
      -value 10
   delaySlider;

   button
      -label "Blink"
      -width 70
      -command "blinkGetInfo"
   button;

      setParent ..;

   showWindow blinkWindow;

}

blinkWindow;
```


TERWORLD

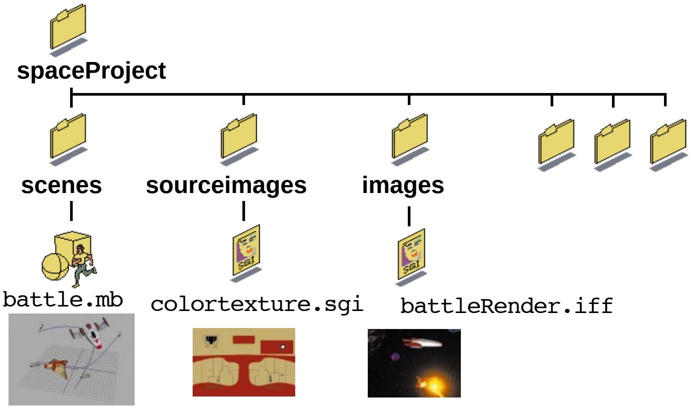
Appendix

APPENDIX A

A File Management

When you are working with Maya, you need to create and save your scene files into directories on your hard drive. To help you save and open these files, Maya lets you create project directories where your files can be saved in an easy-to-organize location. You can also work with other files in other directories and even on other computers. By setting up your projects to work with these other directories, you can work in any number of ways.

In this chapter, you will look at different ways in which scenes can be managed using projects.



A sample project directory

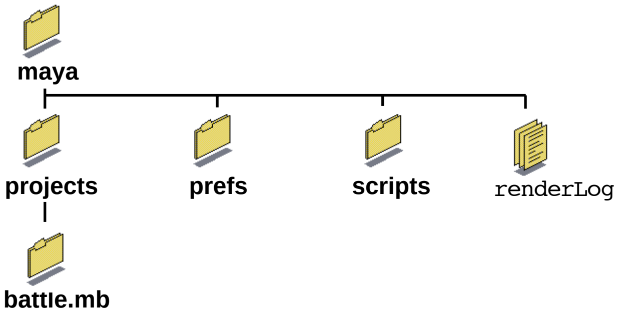
In this appendix, you will learn the following:

- Where your *maya* directory is and what is in it
- How to work with your *prefs* directory
- How to create and edit project directories
- How to work with reference files
- How to share files in a large project
- How Maya works with data transfer file types

The Maya directory

When you first launch Maya, the *maya* directory is placed on your hard drive in your home directory, on IRIX or in your `windowsNT/profiles/username` directory on NT. Each user gets its own *maya* directory. Here you will find some directories and text files. Listed below are some of the directories and files that you will find in the *maya* directory:

- *project* - this directory contains your project directories which in turn contain all of your working files.
- *prefs* - this directory contains system configuration information that saves your Maya settings when you quit Maya.
- *scripts* - this directory gives you a place to put MEL scripts so that commands can be easily sourced from within Maya. You used this directory in Lesson 23 when you saved the *blink.mel* file used to set keys on Salty's eyes.
- *renderLog* - this file contains information about your renderings including render time, and memory usage.



The Maya directory

The preferences directory

The *prefs* directory contains files that tell Maya how you want your system configured. The various user interface settings you work with in Maya such as hotkeys and shelf layouts are recorded here.

How to reset your preferences (safely)

If you want to return Maya to its default settings, you can simply delete the *prefs* directory. Before you delete it, you should rename the existing *prefs* directory in case you need any of your previous settings. This is especially important if you want to work on another person's computer.

If you were to delete their prefs directory, then they could lose their personalized user interface settings for good. To safely reset your preferences, complete the following steps:

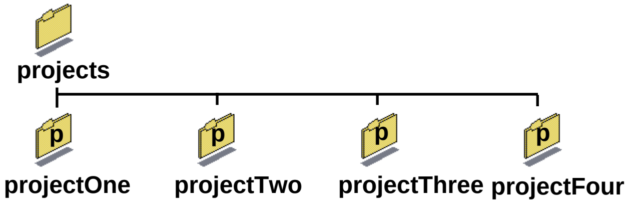
- In your *maya* directory, open your *2.0* directory then rename your *prefs* directory to *old_prefs*.

  This preserves the existing settings.
- If you used a previous version of Maya then do the same for that version. Maya 1.0 preference files will be in your *maya* directory and in your *maya/1.5* directory for maya 1.5.
- Relaunch Maya.

  A new *prefs* directory is created automatically. You can then refer to the *old_prefs* if you want to retrieve one of the old files.

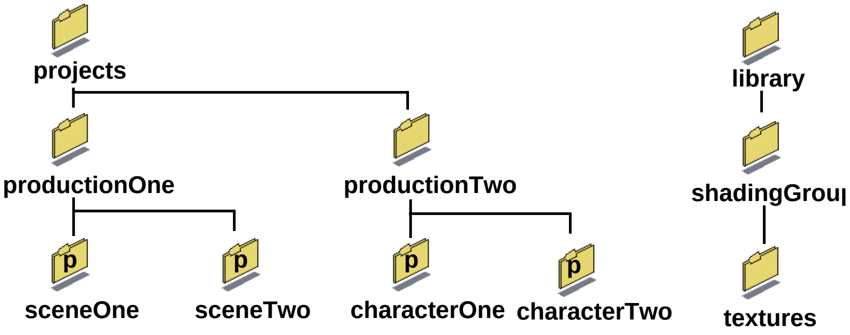
The projects directory

The *projects* directory is where you will store your Maya work. Maya files, source images and final renderings can be stored in this directory. For a simple project directory, you could imagine having several default projects (labelled with a p) set up for each of your animations.



A simple project directory set-up

You do not have to set up your projects in the projects directory. As your projects become more complex, you may want to set up other directory structures in which you can place your project directories. In the following example, some projects have been placed within production-focused directories within the default projects directory, while a texture library has been set up in another location.



A more complex project directory setup

Creating a new project

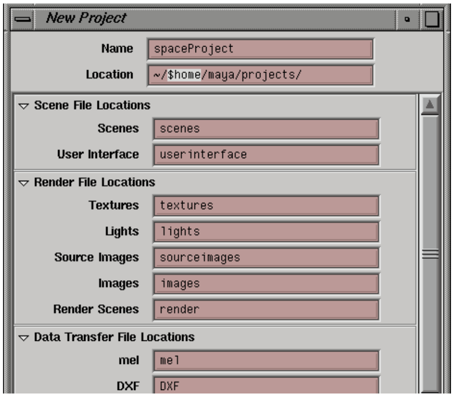
When you create a project, you set up the location of the project and the names of the subdirectories. These subdirectories let you further break down a project so that the parts can be easily retrieved by yourself and other members of your production team, however big or small.

To create a new project:

- From the **File** menu, select **Project** → **New...**

  The new project window appears with fields available for different types of subdirectories.

- Click on the **Use Defaults** button to enter the Maya defaults into the fields.
- Delete text from any field where you don't want a subdirectory.



New Project

Name	spaceProject
Location	~/$home/maya/projects/

▽ Scene File Locations

Scenes	scenes
User Interface	userinterface

▽ Render File Locations

Textures	textures
Lights	lights
Source Images	sourceimages
Images	images
Render Scenes	render

▽ Data Transfer File Locations

mel	mel
DXF	DXF

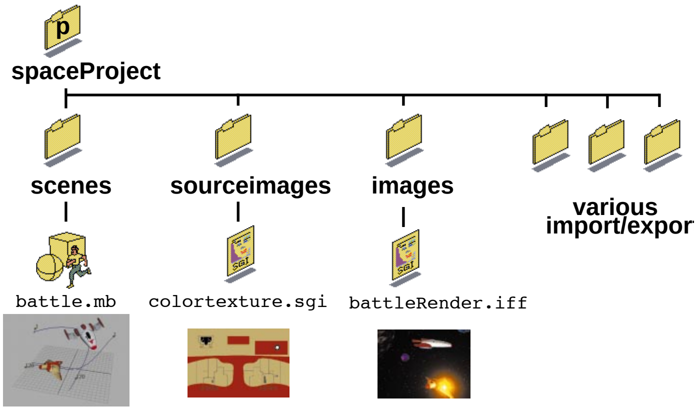
New project directory

The names you enter here will be used when Maya points to a particular directory type – such as the *Images* directory type – when you render. You could name your *Images* directory *artwork*. If you leave a directory type field blank, then no subdirectory is created.

Some of the most important directories are as follows:

- *scenes* - This is where you save your Maya scene files.
- *sourceimages* - This is where you put image files that you want to source as file textures or as image planes.
- *images* - This is where your rendered images go.

In the following diagram, you can see some of the file types. The other project subdirectories offer particular locations primarily for import and export files which you want to associate with your project.



A sample project directory

Setting projects

As you will probably be working with multiple projects at the same time, you need to be able to switch between them. While at any time, you can open any file from any of your projects, it is advisable to only open files when you have set the appropriate project directory. Once you set a project, Maya then points towards the right directories for opening and saving files. For instance, your rendered images go, by default, to the *images* directory of the current project.

To set a different project:

- Select **File** → **Project** → **Set...**

- Choose another project to make your active project. This is now where Maya will look when opening and saving files.

Scene files

Your Maya scenes are saved as either *Maya binary* or *Maya ascii*. The first format is faster for loading and saving files while the second format can be read and edited (if you really know what you are doing) with a basic text editor. If you do not need to know the meaning and syntax of the file format then *Maya binary* is probably good for you.

REFERENCING FILES

In Project Three of this book, you created an animated space scene by referencing two spaceship files. While this seemed to be just an elaborate kind of import, referencing lets you share the referenced scene files with other scenes. Then, if you wanted to update one of the models or change one of the texture maps, all of the referencing scenes would benefit from the changes.

File referencing lets you reference one scene file into another. This lets you have a series of prop and character files that are referenced into other scenes. These props and characters can be shared by other animators working on other scenes which helps streamline the production process.

To create a reference:

- Select **File → Create Reference...**
- Choose a file for referencing.

  This file can be in the current project, in another project or even on a remote computer.
- Press the **Reference** button.

To view the reference:

- Select **File → Reference Editor...**

  This window shows you the hierarchical relationship of the main scene file, referenced files and associated UI files.

  If you click on one of the files, you get its particulars. One important point is the **I/O** (Input/Output).

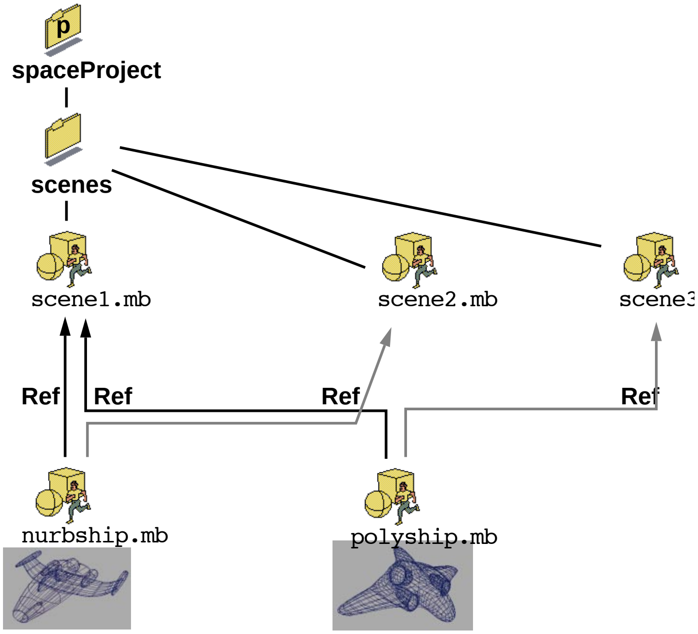
To delete a referenced file:

- Open the **Reference Editor**.

- Select the reference in the editor.
- Select **Edit** → **Remove Reference**.

A single user project with referencing

A single user or a small team can use file referencing to share models, shading groups, lights and cameras between different scenes. Below is an example of the space scene as if it was being worked on by a single user in a single project directory:



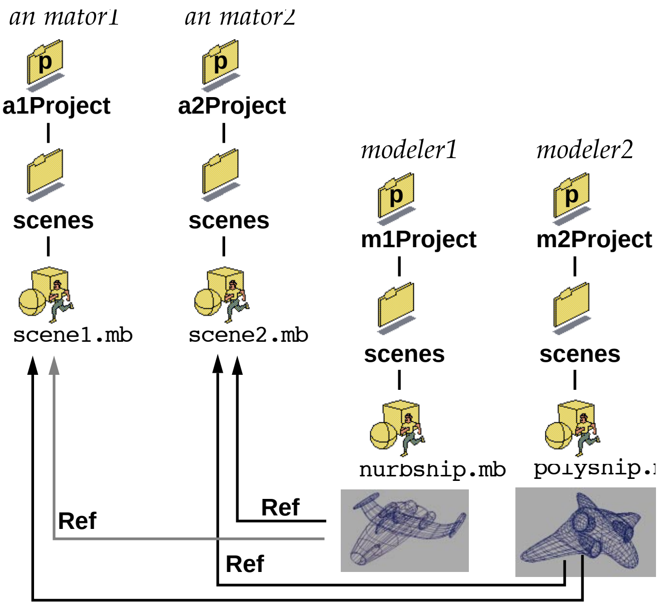
Single user referencing

In this example, the two spaceships are being shared by multiple scenes. Both ships are referenced into *scene1*, while only one of the ships is referenced into the other scenes. The appropriate scenes will automatically update changes made to either the first two ships.

These changes are subject to the *rules of referencing* as outlined at the end of this appendix.

A multiple-user project with referencing

You can also share files amongst different users and different projects. In this case, the files must be more carefully set up in the projects directory so that your project can see the projects of other people.



Multiple-user referencing

After referencing the models, *animator1* and *animator2* must set up their projects to always point at the modeler projects where the referenced files are kept. By default, Maya only looks for files in the current project's subdirectories. To look elsewhere, you must point to the other projects:

How to set up the project for animator1 and animator2

- Select **File** → **Project** → **Edit Current**.
- Under **Scene File Locations**, change **Scenes** to read the following:

```
scenes: /home/modeler1/maya/projects/a2Project/scenes:
/home/modeler2/maya/projects/a2Project/scenes
```

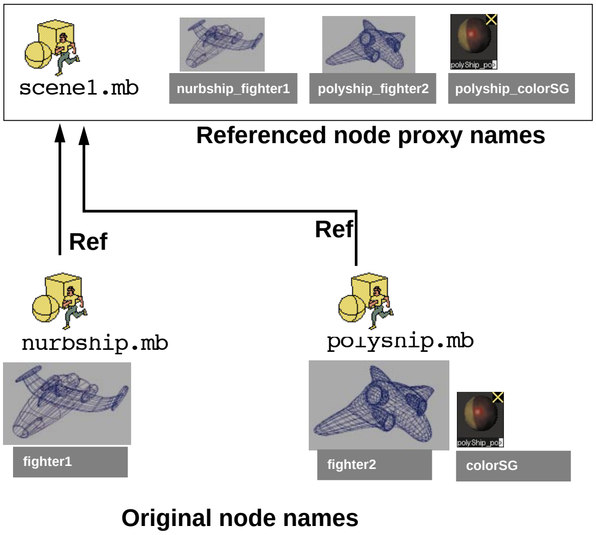
In this case `/home/` represents the home directory of each modeler. Now when the animators open their own scene files, Maya first checks the current project's *scenes* directory for any referenced files, then it will check the modelers' directories. The colon is used to separates directories.

Understanding References

In Maya you work with the dependency graph which is built using *nodes*, *attributes* and *connections*. Both your scene file and any referenced scene files contain these basic building blocks.

When files are referenced, the nodes are placed into the new scene as *Read Only*. Every node and connection from the reference file is placed into the scene with a prefix showing the file's name. In the example shown below, you see that when the file called *nurbship.mb* is referenced into *scene1.mb*, then its *fighter1* group node is given the new name of *nurbship_fighter1*.

This renamed node is actually a new node in the parent scene file that acts as a *proxy* for the referenced nodes. Any modifications made to the attributes on these proxy nodes are stored in the *scene1.mb* file and are not sent back to the *nurbship.mb* file.



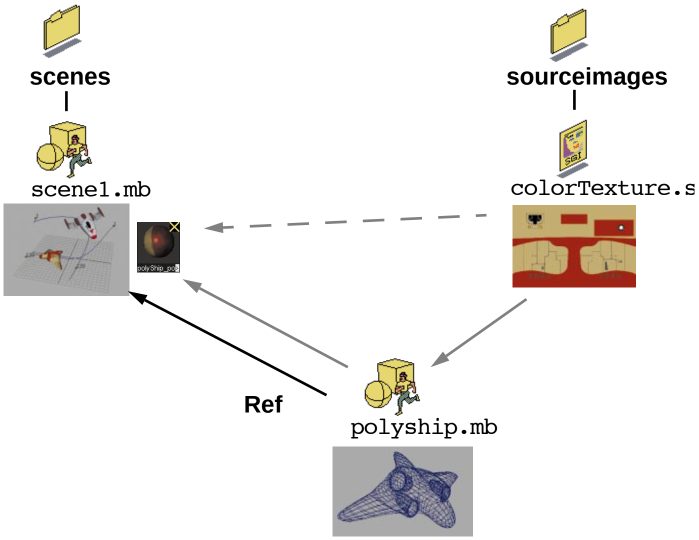
Renaming of referenced nodes

What happens to referenced textures and shaders

When you reference a file, the texture and shading group nodes are treated just like other nodes. They are renamed with a prefix to avoid name clashes and added to the scene file.

Source images that might have been loaded into a file texture would be referenced forward as part of the shading group.

If the file texture image had not been loaded into the original file, it could be added to a file texture node in the *scene1.mb* file and then connected to the shading group. In this case, the texture would not be part of the original *polyship.mb* file and therefore would not be shared with other people sharing *polyship.mb*.



Sourceimages and shading groups being referenced

Importing vs. Referencing

To get another scene to be part of your scene can be accomplished in two different ways. You can import the scene in order to *merge* the files or you can create a reference as you have learned in this appendix. If you want to share the files amongst many scenes then you would want to reference otherwise importing should work fine.

Exporting parts of a file with reference

If you have a file that contains a character, camera, light or other object that you want to share with other users, you can export the file as a reference. This means that it is essentially removed from your scene file then saved as a separate referenced file. Other people can now reference it into their scenes.

To export as a reference:

- Select the pieces that you want to export.
- Select **File** → **Export Selection** - ❐.

- In the option window, set **Keep Only a Reference** to **On**.
- Click **Export Selection**. Now the selected portions of the scene are placed into a new scene file that is automatically referenced.

Making and breaking reference connections

When you are referencing files, you are placing proxy nodes from those files into your scene. As you set keys, move objects and change attributes to those proxy nodes, you are making links to the attributes on these nodes. With reference files these connections link the proxies to nodes that sit in the scene file.

It is important to remember that all you have to do is change one attribute on a node and Maya records a connection into your scene file. For instance, if you reference in a file that contains a model of a chair and then you move the chair's root node, that node has been connected. You cannot rename or group/ungroup this node in the reference file or the connection will be broken. You can rename referenced nodes below the chair's root node because these nodes weren't connected.

Once these connections take place, you must avoid *breaking* the connections. Below are some tips for working with these files.

The rules of referencing

- If you save your files as *Maya ascii* then you can open your scene file in a text editor. This will then allow you to see what kinds of connections are made to your referenced nodes. Also, the actual reference is listed at the top of the file.
- Referenced files should not be renamed or else the scene file will not be able to find it.
- If you want to move a referenced file to another directory, make sure to set up the project to look for that directory. See *How to set up the project for animator1 and animator2* earlier in this chapter.
- Nodes in referenced files should not be renamed or regrouped unless you are positive that the scene file has not made any connections. If you set keys or even change the value of the proxy version of a node, then the original node must not be renamed or regrouped or else connections will be broken.
- If you want to replace a referenced file with another file with the same name, remember that the new file must contain all of the nodes (named properly) that are connected in the scene file.

- If you want to apply a deformer to a referenced surface, go back to the original referenced scene file and apply the deformer there. Since deformers link to CV names and numbers, if you were to later attempt to create a new surface in the referenced file, the deformer connections would break.
- Some expressions that use MEL command statements can give your problems. For example, *Ball.ty = 5* would definitely work whereas *setAttr Ball.tx = 5* will not work.
- When you are writing MEL scripts for projects with referenced files, remember that the object names will have the prefix. Efficient use of variables should help you coordinate scripts in a reference file scenario.

Index

Notes

Notes

NOTES

Notes

NOTES

Notes

NOTES

Notes

Notes

Notes

NOTES

Maya Training resources

The **Learning Maya** book offers you self-study training right out of the box. You can guide yourself through several projects to help you learn the interface, workflows and functionality found in Maya.

To continue learning Maya in an instructor-led environment, courses are available that offer more advanced training in various Maya concepts and workflows. For information on all Alias|Wavefront training classes, please visit our website at **www.aliaswavefront.com** and choose the Training Express link.

If you would like to register for a training session, or if you do not have access to the web, you can contact your local sales office or call the *Training Information Hotline* at **416 362-8558 x5699**. You can also email us at **training-info@aw.sgi.com** or fax us for information at **416 369-6131**.

Another great Maya learning resource is the Assistant Online. This web site offers an exciting new way to learn as you work directly with digital files and web-based lessons. This web site also includes shaders, MEL scripts, sample files and many How-to lessons. You can access the Assistant Online site at **www.aliaswavefront.com/assistant_online**.