

PIXELS 3D 4.5 User Guide

By Mark and Ananda Stevens

Software created by Andrew Bryant

Programmers: Jimy Arndt, Toby Miller, Willis Morse, Walter Hunt, Olivier Jean-Marie

Product Manager: James Ricks, and Steve Yatson

Engineering and Customer Support: Mark Stevens

PiXELS 3D 4.5 User Guide
2002 Pixels Digital, Inc. All Rights Reserved.

A notice of protection under copyright, trade secret, and contract law may be found at the back of this manual and on separate documents packaged with the PiXELS 3D 4.5 software.

The use and copying of the PiXELS 3D 4.5 User Guide and SOFTWARE are subject to the License Agreement included with this software. Any other use is prohibited.

Contents

Introduction	1
What's New in PiXELS3D 4.0.....	1
What's New in PiXELS 3D 4.1	3
What's New in PiXELS 3D 4.5	3
Pixels Web Site	4
Getting Started	7
System Requirements	7
Minimum	7
Recommended	7
Documentation	7
Conventions	7
Installation	9
Main Window	13
Default Layout	13
Panning or Scrolling	14
View Panes	14
View Pane Controls	15
View Pane Hot Keys	21
View Pane contextual menu	22
Function Keys	26
Menu Bar	26
Favorites Bar	27
Layout Selector	27
Modeling Layout	28
Shading Layout	29
Animation Layout	29
Layout Manager	30
The Basics	33
Launch PiXELS3D	33
Create a cube	33
Move the View	34
Change the Camera View	35

Move the Camera	35
Rotate the Cube	36
Undo the Rotation	37
Rotate Around the Y and Z Axis	37
Creating NURBS and Curves	39
Principles	39
Curves	39
Periodic NURBS, U, and V	39
U Rows and V Columns	41
NURBS Primitives	42
Sphere	43
Cylinder	44
Cone	46
Mesh	49
Torus	51
Cube	53
Curves	55
Freehand	55
Arc	56
Circle	57
Square	58
Helix	59
Star	60
nGon	61
Gear	62
Constructed NURBS Objects	65
Lathe	66
Loft	67
Sweep	68
Extrude	69
Flying the Flag	71
Modeling the Flag	71
Shading the Flag	73
Highlighting and Shadows	75
Orientation	77

Casting Shadows	78
Specularity	80
NURBS Modeling	83
FreeForm	83
Mode	83
Flex	85
Shape Curve	85
Modifiers	88
Twist	89
Taper	89
Wave	90
Disrupt	91
Flip	92
Shear	92
Quantize	93
Attractor	94
Bulge	95
Deformers	95
Conversion	95
Editing the NURBS Surface	96
Wrap	101
Curve Control	103
Degree	103
Knots	104
Select	104
Attributes	104
Transformation	105
Render Options	105
NURBS	105
Display	105
Morph Targets	105
Modeling a Molly	107
Modeling a Face	119
Create the Nose	119
Create A NURBS Cylinder	121

Pinch the End of the Cylinder	122
Pull the End of the Cylinder	125
Expand the Middle of the Cylinder	125
Save Your Work	126
Create the Head	127
Create a NURBS Sphere	127
Move and Scale the Sphere	128
Connect the Sphere	129
Save Your Work	130
Reshape the Head	131
Add Eye Sockets	133
Save Your Work	135
Add Eyebrows	136
Create a Cylinder	136
Shape the Eyebrow	136
Create the Other Eye	138
Save Your Work	139
Add Eyes	139
Add Beard	140
Add Hat and Hair	142
The Finished Wizard	144
LogoMaker	147
Getting Started	147
Creating the Text	147
Importing the Text	148
Drawing Splines in PiXELS 3D	149
Working with the Text	151
Object Attributes	151
Polygonal Objects	157
Gallery of Objects	157
2D	157
3D	158

Metaballs	161
Shading	165
Shaders vs. Materials	165
Subsurface Scatter	167
Volumetrics	167
Shading Interface	167
ShaderManager	167
TextureManager	168
ShaderMaker	169
Object Node	170
object input parameters	170
Nodes	172
Color Multiplication	174
Connecting Nodes	176
Surface Nodes	178
Editing Inputs	181
Ka, Kd, and Ks	182
Global Coordinates	182
2D Surface Textures	184
Gallery of Nodes	186
Ambient Occlusion	186
Atmosphere	187
Blur	190
Brick	191
Cellular	192
Clay	198
Clouds	199
Comp (Component)	200
Color Math	200
Color Ramp	202
Diffuse	202
fBm	204
Fractal Sum	206
Frame Number	207
Gate	209
Granite	211

Image Map	212
Invert	214
Marble	215
Math Functions	216
N	219
P	220
Posterize	220
ProbeLight	222
Ramp	222
Reflect	222
Refract	223
S	223
Skin	224
SM3 Surface	224
Spots	225
Surface	225
T	226
Tile	227
Toon	227
Turbulence	228
User Defined	229
Velvet	230
Wave2D	230
Wave3D	231
Weave	232
Wood	234
Real Rock	235
Granite	235
File Types	241
Scene Files	241
Polygonal Model Files	241
Render Files	242
Shader Files	243
ShaderManager	243
ShaderMaker	244
Texture Files	244

TextureManager	244
ShaderMaker Image Map Node	244
Tag Maps	244
Rendered Files	245
Still Images	245
Animations	246
Creating a Texture	249
Cameras and Composition	253
Camera Parameters	253
Using Multiple Cameras	254
Director camera	254
View camera	255
Composition	255
The Rule of Thirds	255
Drawing Attention to the Subject	256
Lights and Lighting	259
Light Parameters	259
Rendering	265
Tempest	265
Sky Dome	266
General Render Options	267
Modeling Olympus Mons	269
Terrain From Shaders	270
Image Map Node	270
Displacement	271
Displacement Bounds	272
Making a Negative with a Blender	273
Raytracing	275
Reflection	275
Specular Highlights	276
Background	278
Refraction	279
Air Bubble	280

IOR Values	280
Raytraced Shadows	281
Area Lights	282
Keyframes and fCurves	283
Timeline	283
Keyframes	283
Automatic Keyframes	284
Manual Keyframes	285
fCurves	286
Constraints	288
Path	288
Look At	290
Orbit	291
Snap	291
Limits	291
Tutorial: Roller Coaster	293
Particle Animation	303
Particle Systems	303
Emitters	303
Display and Appearance	303
Life and Death	307
Deviation	308
Dynamics	308
Gravity	309
Wind	310
Floor	311
Forces	311
Accelerator	312
Friction	312
Gravity	313
Collision	313
Vortex	314
Turbulence	315
Death	315
Life	316

Wind	316
Character Animation Tools	319
Spline Deformers	319
Lattices	320
Bones and IK	322
Prepare Your Model	322
Create the IK System	324
Position the IK System	325
Attach the Bones	327
Animate the System	329
IK Modifiers	329
Add Bone Modifiers	330
Animate the System	331
Morph Targets	331
Lofted Blends	333
PixelScript	339
Learning Tcl	339
Creating/Editing Scripts	340
Expressions	340
Menu Commands	341
The Scripts Menu	341
Reset	341
Fill Colors	341
Toggle Black	341
Pencil Test	341
Interface Appearance	343
Legal Notice	347
PiXELS3D Version 4 End User License Agreement (EULA)	347

Chapter I. Introduction

Welcome and congratulations for purchasing PiXELS 3D 4.5! This version continues our ongoing commitment to deliver incredibly powerful professional grade products at affordable prices, allowing students and independent Immakers to experience high-end functionality without the high-end cost.

PiXELS 3D is more than just software - It's a community! We at pixels invite and encourage you to participate in our on-line community, which includes a message board, web ring, contests, and links to various Web pages from the Pixels site (<http://www.pixels.net>). Further, PiXELS 3D 4.5 continues our tradition of soliciting and responding to customer feedback. Many of its features are the direct result of feedback and feature requests. Want to help shape PiXELS 3D development? Let us know what features you would like to see in future versions!

If you need to reach us directly, you may do so at: Pixels Digital, Inc., 9921 Carmel Mountain Road # 312, San Diego, CA 92129 or via e-mail to support@pixels.net.

Power to the People!

Andrew Bryant, President, CEO, and Founder

Pixels Digital, Inc.

What's New in PiXELS3D 4.0

PiXELS 3D 4.0 builds on the tradition of its predecessors by combining amazing power and usability into a single easy to learn application that will truly unleash your creative juices!

PiXELS 3D 4.0 has been completely redesigned from the ground up to be even better than before.

Shadermaker integration. PiXELS 3D 4.0 includes the powerful ShaderMaker Pro (previously a separate product) directly within the application.

New shader nodes. PiXELS 3D 4.0 includes new shader nodes to give you more flexibility in creating new shaders.

Tempest renderer. This renderer gives you similar power to the renderers used by major motion picture studios. The Tempest renderer features:

- Micropolygon displacement

- 3D motion blur

- Low memory requirements

- Perfectly smooth surfaces - with no visible polygons

- Programmable shaders

Stackable, non-destructive deformers and modifiers. Add as many modifiers and deformers as you like to a given object to create totally unique effects.

100% customizable, Open3D™ architecture. Power and flexibility are PiXELS 3D 4.0's hallmarks. This gives developers new freedom to create more scripts and plug-ins than ever before.

Super-fast OpenGL previews. Take advantage of the latest video acceleration technology to spend more time creating and less time waiting to see how your work is turning out.

Clean, easy-to-use interface. The PiXELS 3D 4.0 interface has been completely redesigned to be totally modular, customizable, and very user-friendly.

Solid, NURBS, polygon, and metaball modeling. PiXELS 3D 4.0 allows you to mix and match different kinds of modeling in a single scene.

Every parameter can be animated. The all-new Attribute Manager allows you to animate any aspect of your scene.

What's New in PiXELS 3D 4.1

Since PiXELS 3D 4.0 was first released the following features have been added.

Raytracing. Raytraced reflection, refraction, and shadows have been added.

Inverse Kinematics. Completely revised inverse kinematics chains that allow you to more easily create animatable articulated objects.

OS X native. PiXELS 3D is now Mac OS X native.

Faster rendering. The rendering engine has been improved for much faster rendering.

Software Developer Kit. Now in addition to writing scripts developers can add their own features to PiXELS 3D.

What's New in PiXELS 3D 4.5

LogoMaker. Import logos from Adobe Illustrator files or draw them directly in PiXELS 3D. LogoMaker lets you animate extruded 3D text and logos in your scenes.

RigMaker. Inverse Kinematic bone systems just got a whole lot easier with the introduction of RigMaker. No more mess-

ing with influences or envelopes. Just create the bones, add them as a modifier and they work.

Ambient Occlusion. Add realistic ambient light without the long render times associated with global illumination solutions like radiosity.

Sky Dome. Simulate real out-door lighting conditions by using photographs or drawings of the sky.

Area Lights. Turn any light into an area light so that it casts soft, realistic raytraced shadows.

New Import/Export Options. Import and export models in 3ds, lwo, dxf, obj, and rib formats.

New Shader Nodes. Add even more power and flexibility to ShaderMaker Pro with new shader nodes including cellular noise, wave 2D, wave 3D, gate, weave, and derivative nodes.

New Modifiers. Includes two new modifiers: bulge and attractor. Also includes improvements to the wave modifier.

Distance Constraint. Connect a chain of objects one to the other so they follow each other around like a chain of balloons or a string of connected train cars.

Quick Renders. Produces small, extra-fast renders so you can get a quick view of what the final render will look like.

Point Info Dialog. Information about individual points so you can fine-tune the influence of bones.

Pixels Web Site

Visit us regularly at <http://www.pixels.net> for new product information, technical support, upgrade pricing and promotions. The Pixels web site hosts a message board where users

regularly give and get advice and show off their work. In addition, there are links to individual user sites where you can find all sorts of PiXELS 3D related information.

Chapter 2. Getting Started

System Requirements

Following are the minimum and recommend system requirements:

Minimum

Power Macintosh

Mac OS X (PiXELS 3D 4.0 through 4.1.2 require also support OS 9.x. An OS 9 version of 4.5 will be available soon.)

192MB RAM

60MB free hard disk space

QuickTime 4

Recommended

G4 processor, 1.25GHz

Mac OS X 10.2.x

1.5GB RAM

CD-R or CD-RW drive (for archiving/ backing up your work)

2+ GB free hard disk space (for storing textures, rendered output, etc.)

QuickTime 5

Video card with OpenGL support and 128Mb Video Ram

Documentation

Conventions

PiXELS 3D documentation uses several notational conventions to present information of special importance:

WARNING! Warnings indicate items which could cause you to lose data if not addressed.

Note: Notes describe important things to keep in mind, including tips, tricks, and reminders. Ignoring notes will not cause any ill effects.

Lists of items, points to consider, or procedures that do not need to be performed in a specific order appear in bulleted lists:

Item 1

Item 2

Procedures that must be followed in a specific order appear in numbered steps:

1. Perform this step first
2. Perform this step second

Often a numbered list is preceded by a description of the task that these steps will accomplish:

To do something...

1. Do this.
2. Then do this.

Buttons, menus, window titles, other interface text, keyboard keys, and scripting commands are shown in a different font from the body text. For example:

Press the **F9** key to switch to **Object** mode.

Sub-menus are separated from parent menus with a >.

From the menus select **Edit NURBS > Close Ends > Sharp**.

Locations in the **Attribute Manager** are described in a similar manner. For example, **Image Setup** under **Render Options** in the

Attribute Manager is sometimes referred to as **Render Options > Image Setup** in the Attribute Manager.

Example scripts and other code appears in a fixed-pitch font, for example:

```
COMMAND < FLAGS> < PARAMETERS> < ARGUMENTS>
```

Technical terms are *emphasized* near their definitions.

Hyperlinks to web pages, e-mail addresses, and definitions of technical terms are underlined. For example:

Go to the Pixels web page at <http://www.pixels.net>.

Installation

To install PiXELS 3D 4.5...

1. Download the compressed program from the web site (www.pixels.net/download/_____).
2. Uncompress the **gz** file. You can use the freely available Aladdin StuffIt Expander (www.aladdinsys.com) or you can use the OS X command line utility **gunzip**. (The file may uncompress automatically.)
3. Open the resulting disk image (**dmg**) file. (It may open automatically.)
4. Double-click on **Install PiXELS3D**.
5. The first time you launch PiXELS 3D, a **Registration** dialog opens. You should have received a serial number via e-mail. Enter your name and serial number as shown in the e-mail. If you have purchased the product and have not received a serial number, send an e-mail to support@pixels.net.

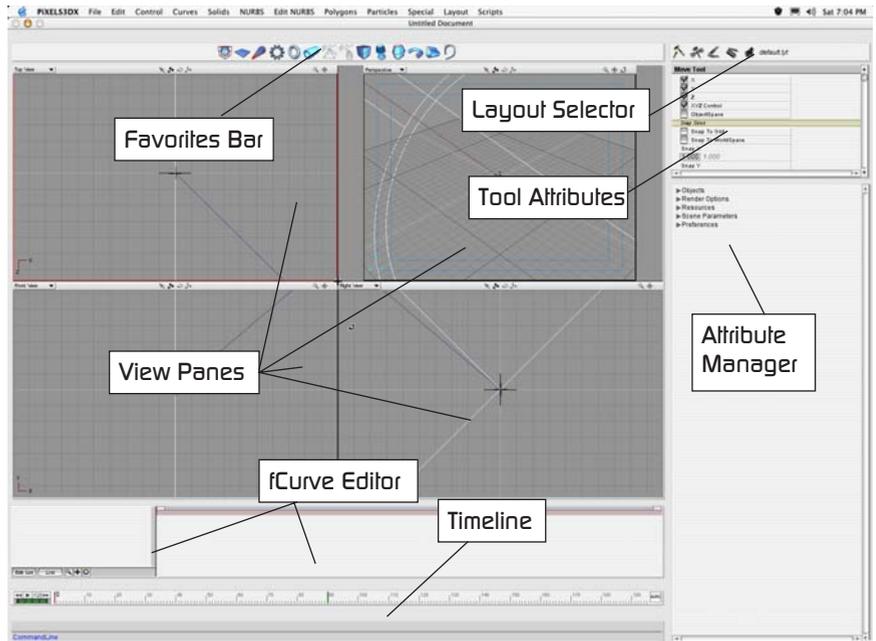
Chapter 3. Main Window

When you start PiXELS for the first time, the main window fills the screen. This window is initially arranged in the **Default Screen Layout**.

Default Layout

This arrangement contains four **View Panes**: the **Top View**, **Front View**, **Right View**, and **Camera View**. The **Top**, **Front** and **Right** views present 2D representations of your scene from the corresponding viewpoints. The **Camera View** shows your scene as it looks from the default camera. As you manipulate and edit an object, you can see the changes occur in each window.

Figure 3.1: Default Screen Layout



Along the top of the window, under the main menu bar, there are two rows of icons: the **Favorites Bar** and the **Layout Selector**. The **Favorites Bar** gives you fast access to frequently used tools, while the **Layout Selector** allows you to switch between different screen layouts.

Along the right side of the **Default Layout** there are two lists: the **Tool Attributes** and the **Attribute Manager**. The **Tool Attributes** list allows you to change characteristics of the current tool. The **Attribute Manager** allows you to edit characteristics of anything in your scene.

Below the **View Panes** the **FCurve** editor allows you to control how attributes change when your scene is animated.

The **Timeline** allows you play a pencil test of the animation or to move from one animation frame to another. The current animation frame is indicated by the numbers on the left and a red marker on the **Timeline**.

Panning or Scrolling

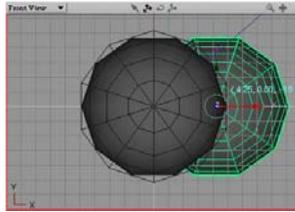
To pan across the view in any window...

Hold down the space bar and drag in the window view with your mouse. On windows that have a scroll bar, you can also use that to move the view.

View Panes

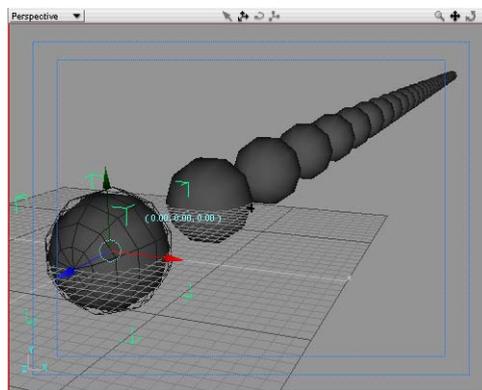
In the default four pane views, three of the views show what are called *Orthogonal Projections*. These are a little like architectural drawings of your scene. In an orthogonal projection objects remain the same apparent size as they move closer to or further from the viewer.

Figure 3.2: Orthogonal Projection



The fourth default view, the **Perspective** view, shows a *Perspective Projection* from the current camera. In a perspective projection objects look smaller as they move away from the camera.

Figure 3.3: Perspective Projection



View Pane Controls

Around each **View Pane** there are several controls for changing the view and editing objects in the view.

 **Sizing.** Any of the four viewing panes can be resized by dragging the + sign that is located where all four viewing panes meet in the center of the interface.

To return to the starting 4 pane size...

1. Switch to a full view by selecting **Expand to Full View** from a **View** menu.

2. Switch back to a 4 pane view by selecting **4 Pane View** from the **View** menu.



View menu. The options on this menu let you change what is displayed in the **View Pane**.

You can change the pane to show a different view, you can toggle between a four pane view and a one pane view, you can change the display mode, and you can hide and show objects.

Top View. Switches the view so that you are looking along the **Y** axis. Objects that are further from the view have more negative **Y** coordinates. The **Top View** is an orthogonal projection.

Front View. Switches the view so that you are looking along the **Z** axis. Objects that are further from the view have more negative **Z** coordinates. The **Front View** is an orthogonal projection.

Right View. Switches the view so that you are looking along the **X** axis. Objects that are further from the view have more negative **X** coordinates. The **Right View** is an orthogonal projection.

Perspective View. Switches the view so that you are looking from the viewpoint of the current camera. A camera looks toward its **Interest** point. To change the current camera, choose **Select Camera** from the **View** menu. The **Perspective View** is a perspective projection.

Left View. Switches the view so that you are looking along the **X** axis. Objects that are further from the view have more positive **X** coordinates. The **Left View** is an orthogonal projection.

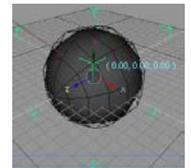
Bottom View. Switches the view so that you are looking along the **Y** axis. Objects that are further from the view

have more positive Y coordinates. The **Bottom View** is an orthogonal projection.

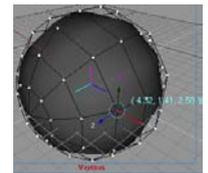
Back View. Switches the view so that you are looking along the Z axis. Objects that are further from the view have more positive Z coordinates. The **Back View** is an orthogonal projection.

Expand to Full View/4 Pane View. Toggles between **Full View** and **4 Pane View**. **Full View** expands the pane so that it fills the whole **View Pane** area.

Objects Mode. In **Objects Mode** you can select and manipulate whole objects. The sphere pictured to the right is selected in **Objects Mode**.



Vertices Mode. In **Vertices Mode** you can select and manipulate individual vertices on an object. A vertex on the sphere pictured to the right is selected in **Vertices Mode**.



Edges Mode [Polygons Only]. In **Edges Mode** you can select and manipulate individual polygon edges. This will be implemented in a future release.

Faces Mode [Polygons Only]. In **Faces Mode** you can select and manipulate individual polygon faces. This will be implemented in a future release.

Select Camera. Allows you to change the current camera. When in **Camera View** you see your scene from the viewpoint of the current camera.

Perspective From Light. Moves the position and orientation of the current camera so that it coincides with the

selected light. After choosing **Perspective From Light**, select a light from the Attribute Manager.

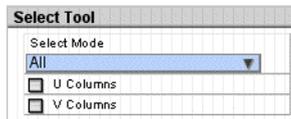
Hide Selected. Hides the selected object(s) so that it is no longer shown and it no longer renders. You can also hide an object by clicking on the second dot to the left of the object name in the Attribute Manager ().

Hide Unselected. Hides everything except the selected object(s).

Show All. Unhides all hidden objects, so everything in your scene is revealed.

 **Select.** Click this icon to switch to the **Select Tool**. With this tool you can select things in the views without worrying about accidentally moving or scaling them.

Figure 3.4: Select Tool Attributes



Select Mode. You can restrict the kind of objects you can select, by changing the **Select Mode**. You can restrict selection to **NURBs** (which include **Solids**), **Polygons**, or **Particles** (which include **Bloppy Systems**).

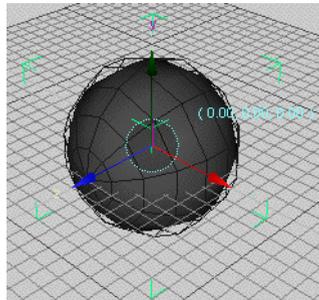
U Rows. When in **Vertices** mode, selects an entire u row on a NURBS object rather than a single vertex. (See **Periodic NURBS, U, and V** on page 39 for more information on u rows.)

V Columns. When in **Vertices** mode, selects an entire v column on a NURBS object rather than a single vertex. (See **Periodic NURBS, U, and V** on page 39 for more information on v columns.)



Move. Click this icon to switch to the **Move Tool**. With this tool you can select and move things. When you select an object a grey circle with colored arrows is shown at the object center. These are the **Movement Handles**.

Figure 3.5: Movement Handles



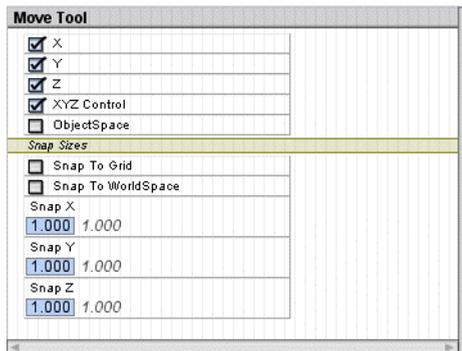
To move the object in any direction,

Click and drag the grey circle.

To move the object along one axis,

Click and drag the colored arrow corresponding to the axis (x is red, y is green, and z is blue).

Figure 3.6: Move Tool Attributes



X. Allows movement along the x axis. Deselect this to prevent movement in the x axis.

Y. Allows movement along the y axis. Deselect this to prevent movement in the y axis.

Z. Allows movement along the z axis. Deselect this to prevent movement in the z axis.

XYZ Control. Enables the grey circular handle where the three colored axis handles meet.

Object Space. Uses the object's original internal coordinate system rather than the world coordinate system. This is particularly useful if you have rotated the object and want to move it along one of its original axes rather than along a world axis.

Snap To Grid. With this enabled, things that you move get shifted to the nearest multiple of the snap distances (**Snap X**, **Snap Y**, and **Snap Z**). Typically the snap distances are all 1 so the thing that you moved gets shifted to the nearest whole number grid coordinate.

Note: You can change the grid size in the Attribute Manager under **Preferences>Drawing Options>3d Views>Perspective Views** and **Preferences>Drawing Options>3d Views>Work Views**. The grid size attributes are called **Size X**, **Size Y**, and **Size Z**.

Snap X. Specifies the snap distance along the x axis.

Snap Y. Specifies the snap distance along the y axis.

Snap Z. Specifies the snap distance along the z axis.



Rotate. Click this icon to switch to the Rotate Tool. With this tool you can select and rotate things. When you select

an object a grey circle with colored rings is shown at the object center. These are the **Rotate Handles**. To rotate the object around the x, y, or z axis, grab and drag the colored x (red), y (green), or z (blue) rotation handle with the mouse. To rotate the object freely, grab and drag the central grey rotation handle with the mouse.



Zoom. Use this icon to zoom into or out of a view. To zoom, click on the icon and drag the mouse up to zoom in or down to zoom out. In the **Camera View**, **Zoom** moves the camera forward and back along the line of sight.

View Pane Hot Keys

In addition to using the controls, you can use the keyboard to manipulate the views.

Pan [Space]. Holding the **Space** bar while dragging the cursor across the view will pan across the view. Panning in the **Camera View** moves the current camera.

Zoom [Option]. Holding the **Option** key while dragging the cursor up and down across the view will zoom in and out. Zooming in the **Camera View** moves the current camera.

No Refresh [Caps Lock]. When **Caps Lock** is on, screen refreshes in the **Views** are turned off. This speeds up performance when working on complex scenes. To refresh while the **Caps Lock** key is on, press **Enter**.

Note: You can also hide objects to speed up performance by clicking the hide button next to the object in the **Attribute Manager**. For example, the blue hide button next to the **Sphere** indicates it is hidden



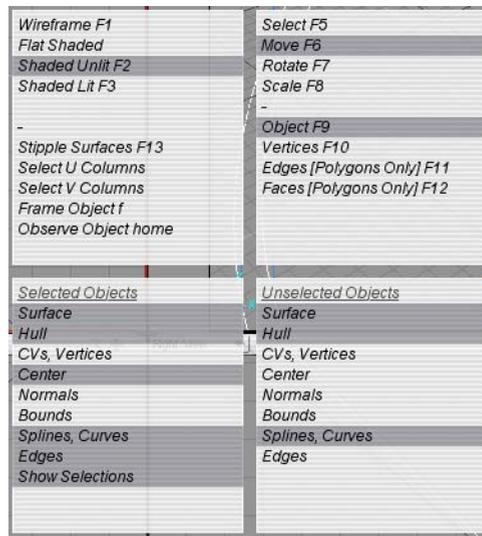
Full Screen [Tab]. The **tab** key toggles between the selected full screen view and the four pane view.

View Pane contextual menu

Force Refresh [Enter]. The Enter key forces the views to refresh even when No Refresh is on.

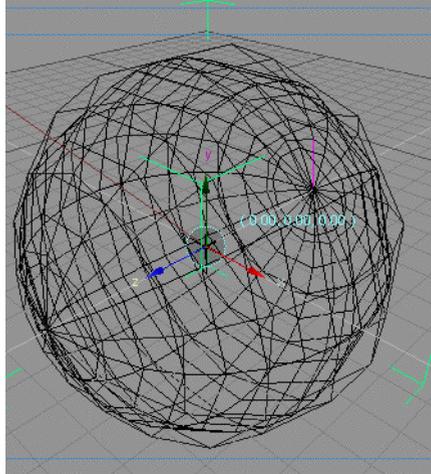
The View Pane contextual menu allows you to change the way objects are displayed in the views. Some of the View Pane contextual menu options are also through the View Pane Controls and some of the options are unique to the View Pane contextual menu.

Figure 3.7: View Pane contextual menu

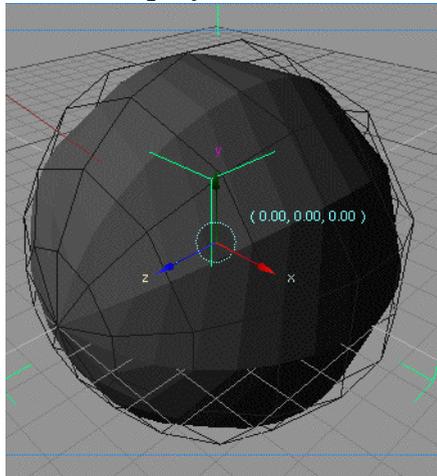


For information about the **Select**, **Move**, **Rotate**, **Scale**, **Object**, **Vertices**, **Edges**, and **Faces** options, see the corresponding entries under **View Pane Controls** on page 15.

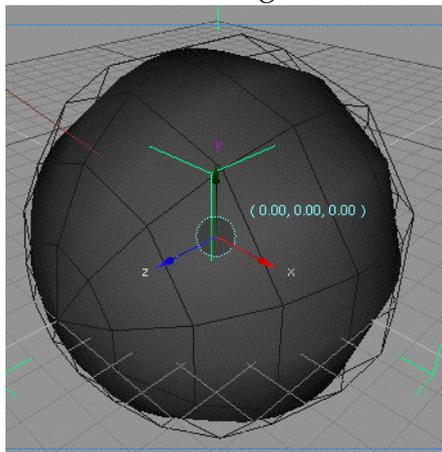
Wireframe. Displays objects as vertices connected by lines. This display allows you to see through everything, but can also become confusing when there are too many vertices.



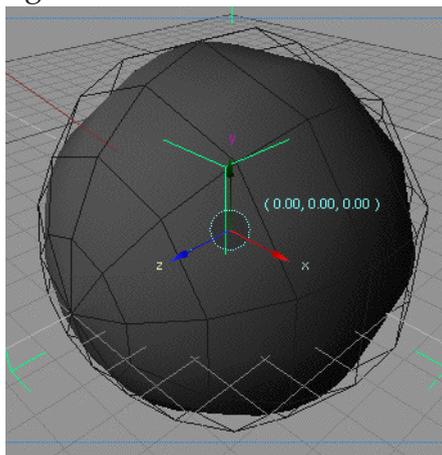
Flat Shaded. Displays objects with faceted surfaces. This is the fastest display mode.



Shaded Unlit. This displays each object as a smoothed surface as if there was a light source at the observer's position.

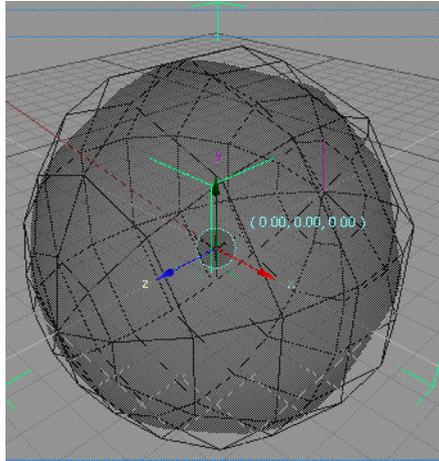


Shaded Lit. This displays each object shaded correctly for the lights in the scene.

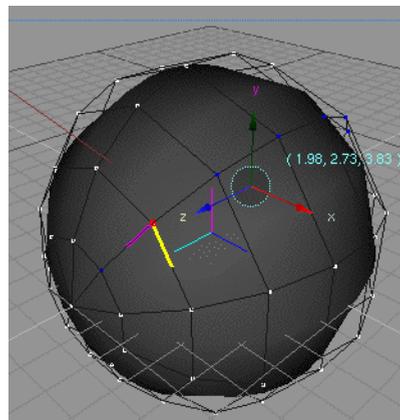


Textured. This will be implemented in a future release.

Stipple Surfaces. Objects are displayed with semi-transparent surfaces so you can see where vertices on the backs of objects are located.

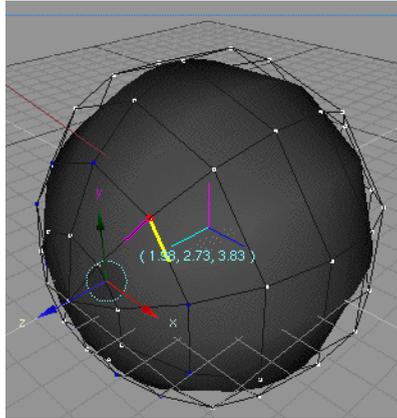


Select U Columns. In this mode, when you select a vertex, all vertices with the same U value are also selected. For a discussion of U and V dimensions, see [Periodic NURBS, U, and V](#) on page 39.



Menu Bar

Select V Columns. In this mode, when you select a vertex, all vertices with the same V value are also selected. For a discussion of U and V dimensions, see [Periodic NURBS, U, and V](#) on page 39.



Frame Object. Centers all the non-camera views on the currently selected object.

Observe Object. Opens the selected object in the Attribute Manager.

Function Keys

Many contextual menu items have function or F-key equivalents.

Menu Bar

Some of the menu items on the menu bar have a plus (+) after the name. This indicates that you will have to respond to one or more prompts in the status bar.

For example, if you select an object and choose **Control>Add Look Constraints+**, you are prompted to **Select Target Object**. In response to the prompt, you need to click on the object your first object is going to look at.

Favorites Bar



To add an item to the Favorites Bar,

Hold down the **Shift** key while clicking on the menu item you want to add. This adds an icon for that menu command to the Favorites Bar.

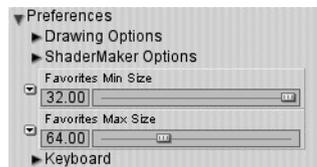
To delete an item from the Favorites Bar,

Hold down the **Shift** key while clicking on the item in the Favorites Bar. This removes the icon.

To adjust the size of the icons...

1. Open the Preferences in the Attribute Manager.
2. Adjust the size sliders. You can adjust the minimum and maximum sizes.

Figure 3.8: Favorites Bar Preferences



Layout Selector



To switch between layouts,

Click one of the layout buttons.

 **Default.** This general purpose layout gives you room to work on modeling and animation at the same time. This layout is described starting on page 13.

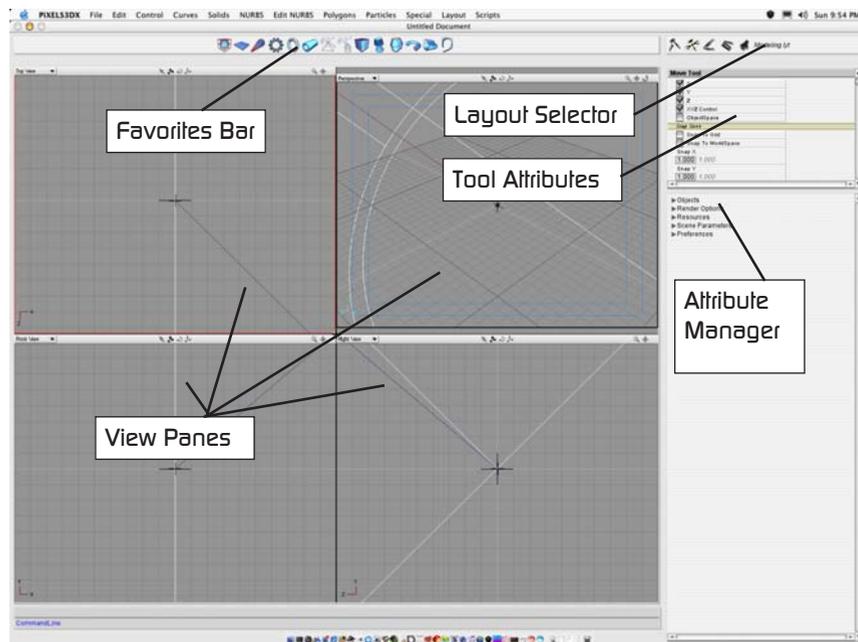
 **Modeling.** This layout omits the Timeline and fCurve Editor to give you more room to model in the views. This layout is described starting on page 28.

-  **Shading.** This layout has a small **Camera View**, has a **Timeline**, and has a large area for constructing shaders. this layout is described starting on page 29. Shader construction is described in **Shading** on page 165.
-  **Animation.** This layout has a large **fCurve Editor**, but smaller views to give you more room to edit the shape of your fCurves (which control the rate of change in your animation). This layout is described starting on page 29.
-  **Layout.** This layout lets you re-arrange the other layouts. To exit **Layout Mode**, choose **Layout>Layout Mode** from the menus. The **Layout Manager** is discussed in more detail starting on page 30.

Modeling Layout

This is very similar to the default layout except the **Timeline** and **fCurve Editor** are removed to make room for larger views.

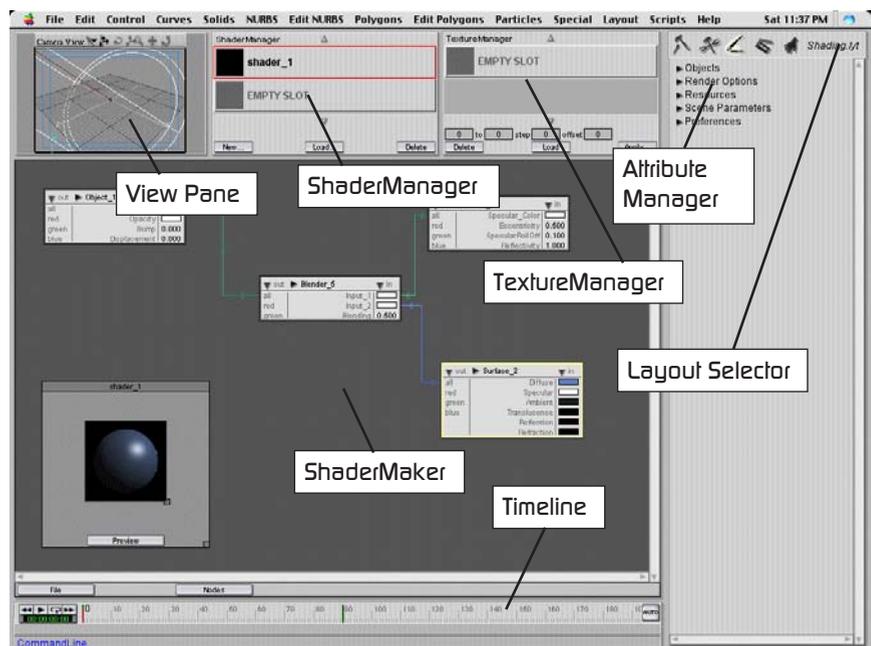
Figure 3.9: Modeling Layout



Shading Layout

This layout gives you the tools you need to construct shaders and assign them to objects. *Shaders* give your objects color, bumpiness, and other attributes that control the way they look when they are rendered. Shaders can be constructed from images listed in the **TextureManager** or from procedural **Nodes** linked together in **ShaderMaker**. For more details about constructing shaders see **Shading** on page 65.

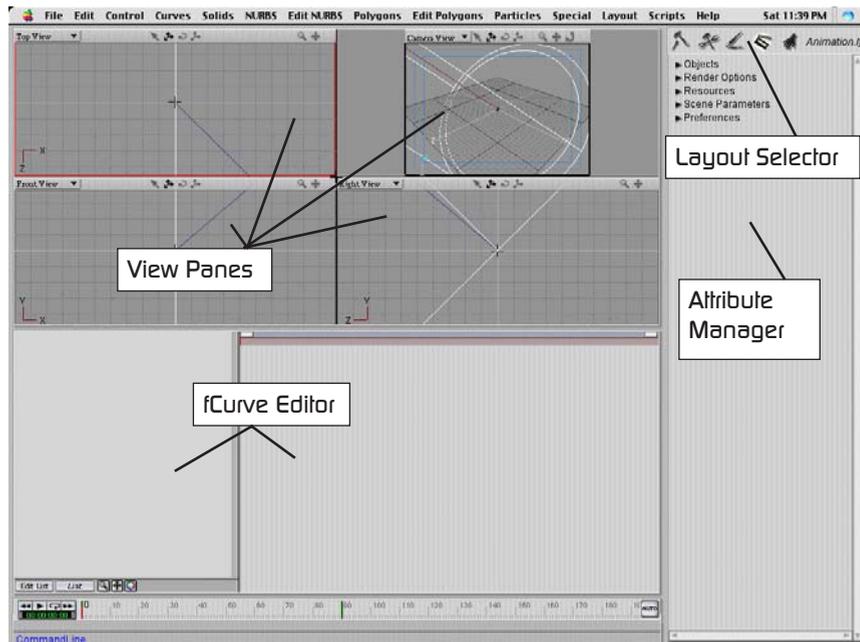
Figure 3.10: Shading Layout



Animation Layout

This is very similar to the default layout except the **Favorites Bar** and **Tool Manager** have been removed to make more room for the **Attribute Manager** and **fCurve Editor**.

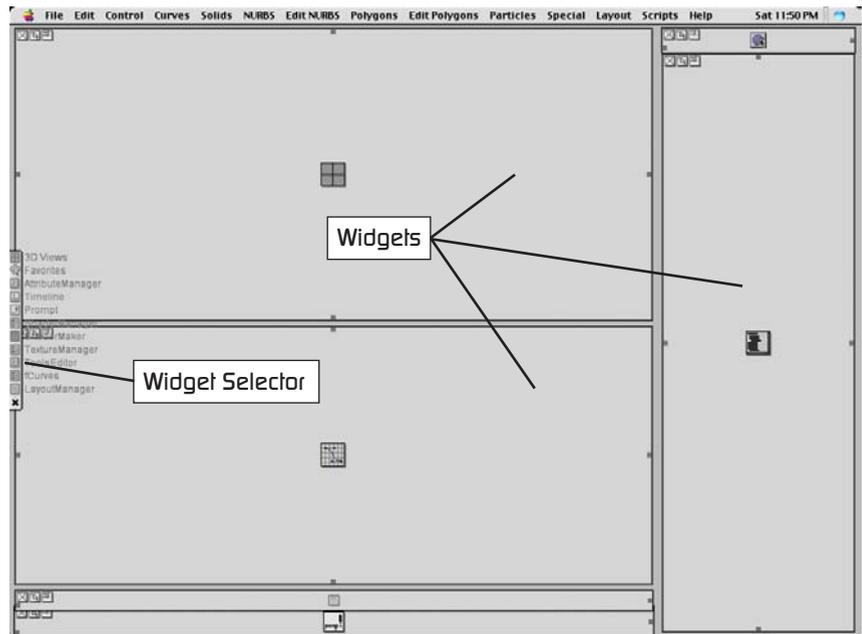
Figure 3.1 I : Animation Layout



Layout Manager

Each interface component in a layout is called a *Widget*. For instance the Attribute Manager, the 3D Views, the fCurve Editor, and the Timeline are all widgets. The Layout Manager lets you rearrange, add, or delete widgets from any of the layouts.

Figure 3.12: Layout Manager



To exit the Layout Manager,

Select **Layout>Layout Mode** from the menus.

To resize a widget,

Click and drag one of the handles along the side of the widget.

To add a widget...

1. Resize the other widgets to make room for the new widget.
2. Click and drag the new widget from the widget selector to the space you made.
3. Adjust the size of the new widget.

To delete a widget,

Click the close box in the widget frame.

To save your changes...

Select **Save Layout** from the **Layout** menu or the **Layout** contextual menu (that is, the menu you get when you **Control-Click** in the window).

To edit a different view,

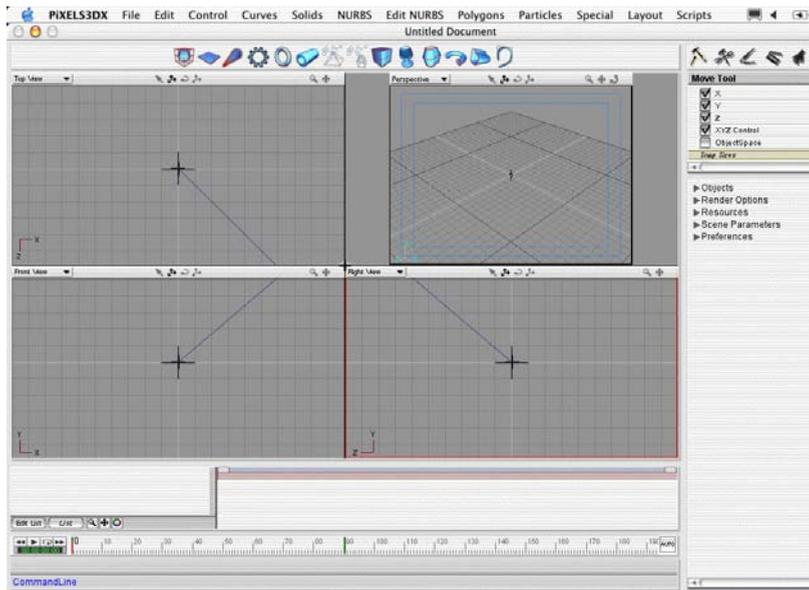
Select the view you want to edit from the **Layout** menu or the **Layout** contextual menu.

Chapter 4. Tutorial: The Basics

Launch PiXELS3D

Double click on the PiXELS 3D icon to launch the application. The PiXELS 3D splash screen will open. After a few seconds the main PiXELS 3D window will open.

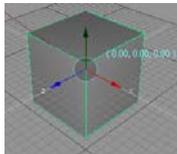
Figure 4.1: Main Window



Create a cube

Select Cube on the Solids menu to create a cube primitive.

Figure 4.2: A Cube



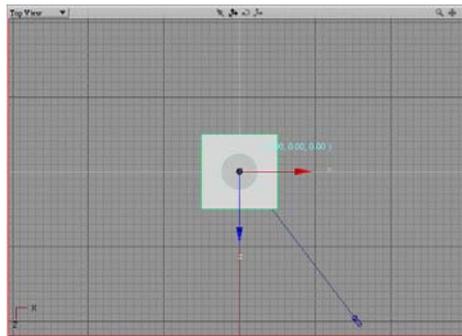
Move the View

Hold down the **Option** key. While it is over any of the views, the cursor now looks like a magnifying glass.

With the **Option** key held down, click and drag the mouse up and down in the **Camera View**. Notice how the view changes.

Using the **Option** key, zoom out (drag down) in the **Top View** until you can see both the cube and the camera. The camera is indicated by a small blue box-shaped icon with a line to the center of the view.

Figure 4.3: Dolly Out in Top View



Instead of using the **Option** key, you can also click and drag on the magnifying glass icon  at the top of any view.

There are keyboard equivalents to all three view icons (there are three view icons in the **Camera View**, but only two in the other views). The view keys are as follows:

 **Zoom or Dolly.** Hold down the **Option** key.

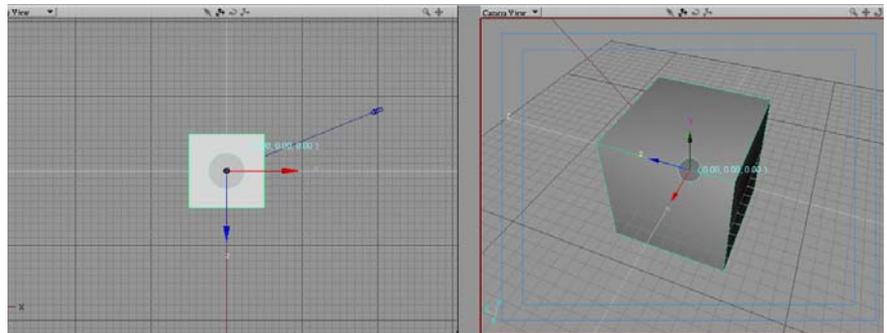
 **Scroll or Pan.** Hold down the **Space** bar.

Change the Camera View

 **Orbit.** Hold down the **Cmd** key and drag in the **Camera View**.

Try orbiting by holding down the **Cmd** key and dragging in the **Camera View**. Notice that the camera changes position in the **Top View**. Any change (dolly, pan, or orbit) to the **Camera View** moves the camera.

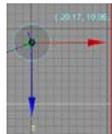
Figure 4.4: The Camera Moves with the Camera View



Move the Camera

Click on the camera in the **Top View**. This selects the camera. The selected object is highlighted in green and has an object handle control positioned at the object's center. In move mode the object handle has arrows at the ends of the **X** (red), **Y** (green), and **Z** (blue) axis handles.

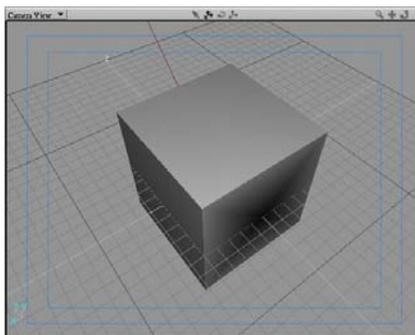
Figure 4.5: Object Handle Control



Rotate the Cube

Click in the grey circle and drag the camera to a different position. Notice that the Camera View changes to reflect the new camera position.

Figure 4.6: New Camera Position



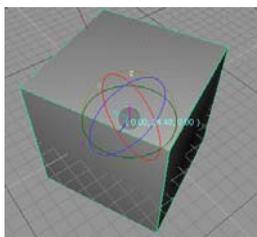
Click on the X (red) axis and drag the camera. Notice that you can only move the camera back and forth along the x-axis. The colored axis handles constrain the current tool to operate along a single axis. The grey circular object handle allows the current tool to operate freely in any axis.

Rotate the Cube

Click on the cube to select it again.

Click on the rotate icon (the curved arrow at the top of a view). The axis handles now look like colored rings. This shows you that you are in object rotation mode.

Figure 4.7: Object Handles in Rotation Mode



In the **Camera View**, click in the grey circular object handle and drag the mouse around. Watch the cube rotate in response to your mouse movements.

Undo the Rotation

From the **Edit** menu select **Undo**. The cube returns to the orientation it had prior to the rotation.

Rotate Around the Y and Z Axis

Click and hold the mouse button down on the Y (green) ring in the **Camera View**.

With the Y (green) handle selected, drag the mouse up and down and from side to side. The ring turns yellow to show that it is selected. The cube rotates around the y-axis in response to the mouse movements.

Now look at the **Tool Attribute** list in the top right portion of the main window (in the default layout). The heading on the list shows the currently selected tool, the **Rotate Tool**.

Click on X and Y in the **Tool Attribute** list to uncheck these axis attributes. Notice that the x and y handles in the views are hidden. Only the Z (blue) handle is visible.

Click and drag the grey circular object handle. The cube only rotates around the z-axis.

Rotate Around the Y and Z Axis

Chapter 5. Creating NURBS and Curves

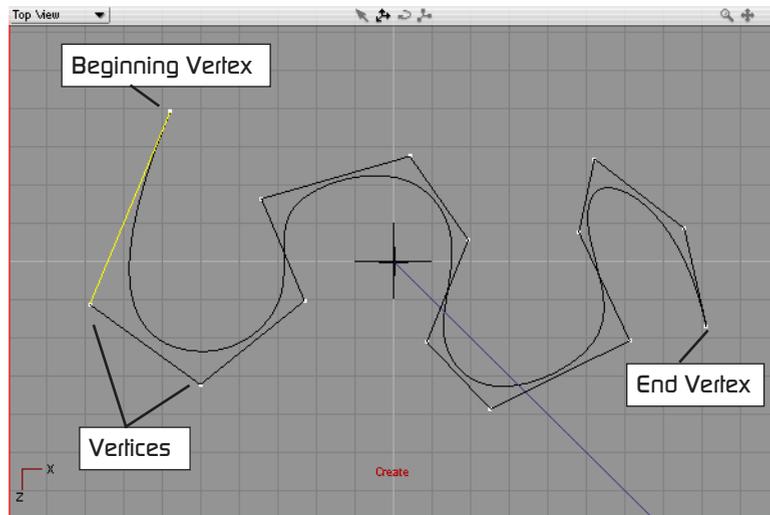
Principles

NURBS stands for Non-Uniform Rational B-Spline. This is a particular kind of curved surface that can be used to model almost anything. *NURBS* are one of the best solutions for creating rounded organic shapes.

Curves

NURBS are formed from a mesh of curves. Curves, in turn, are formed from a sequence of *control vertices*, or points in 3D space.

Figure 5.1: A *NURBS* Curve

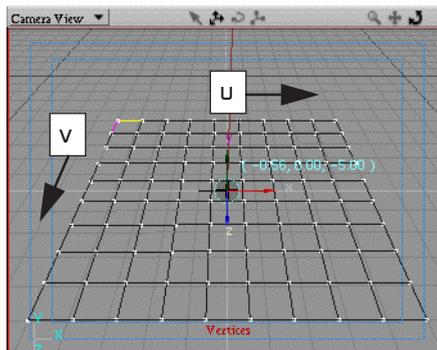


Usually, a curve does not actually pass through any of the control vertices, but does begin and end at the first and last vertex.

Periodic *NURBS*, U, and V

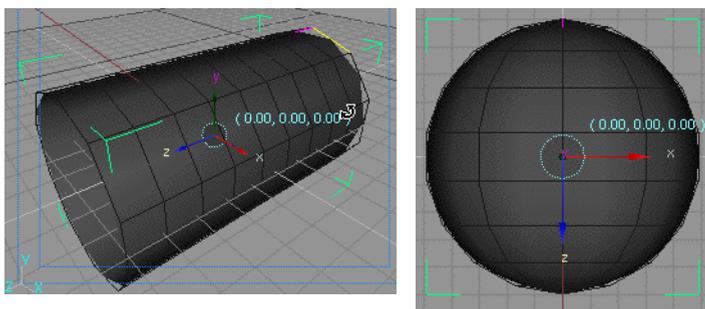
A *NURBS* surface is created from flexible curves. These form a simple mesh.

Figure 5.2: A Mesh



One dimension of the grid is called the u dimension, the other is called the v dimension. The start of the mesh is marked by a yellow line along the u dimension and a magenta line along the v dimension. At this corner point u and v both equal 0. In the opposite corner both u and v equal 1. (In Figure 5.2 the start of the mesh is in the far left corner.)

In some shapes the u dimension is *periodic*. That is, if you travel along a curve in the u dimension, you will return to the point from which you started. Cylinders and spheres are periodic in the u dimension.

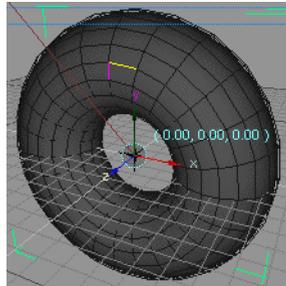
Figure 5.3: Periodic in the u Dimension

Note: A sphere is only periodic in the u dimension. All the vertices at one end of the v dimension coincide at

one pole. They do not join. The vertices at the other end coincide at the other pole.

In some shapes, such as a torus, both the u and v dimensions are periodic.

Figure 5.4: Periodic in Both Dimensions



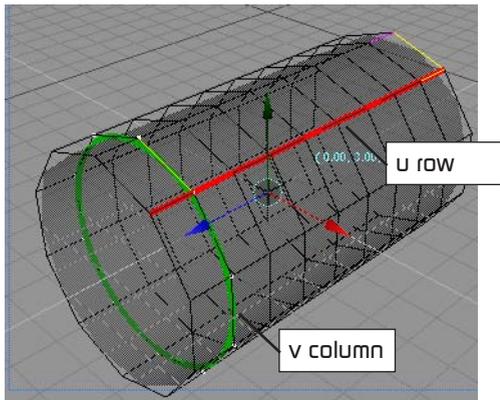
Note: If a shape is periodic in one dimension, but not the other, it is the convention to make the u dimension periodic and leave the v dimension non-periodic.

U Rows and V Columns

Often when talking about vertices, you talk about vertices in a u row or v column. Basically, what this means is that all vertices with the same u coordinate form a u row. All vertices with the same v coordinate form a v column. A u rows or v columns are sometimes also referred to as a u or v *steps*.

In a cylinder, u rows go from one end of the cylinder to the other. v columns form rings going around the cylinder.

Figure 5.5: U Rows and V Columns



NURBS Primitives

NURBS primitives are basic shapes that can be edited using NURBS tools. You can create a NURBS object by selecting an item from the **NURBS** menu or by clicking its icon in the **Favorites Bar**. NURBS icons are indicated by a red frame.

Figure 5.6: NURBS Menu

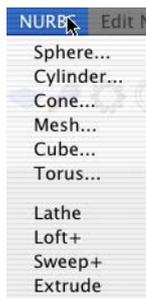
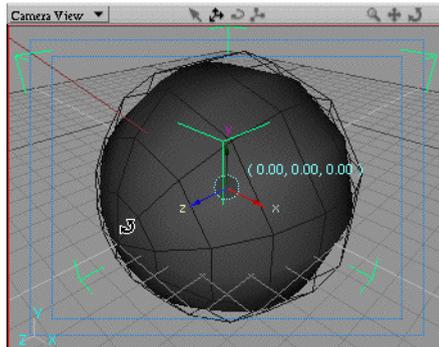


Figure 5.7: Favorites Bar



Many NURBS primitives have options that you can set.

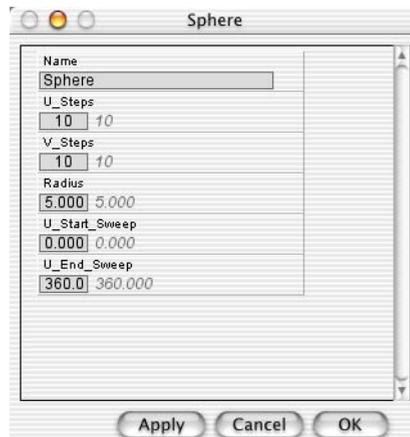
Sphere



To create a sphere...

1. Select **Sphere** from the **NURBS** menu or click the **Sphere** icon in the **Favorites Bar**.
2. When the **Sphere** dialog opens, you can use the default values or change them.

Sphere Dialog



3. Click the **OK** button to create the sphere.

Note: You can use the **Attribute Manager** to change the sphere's attribute values later.

Sphere Dialog Options. The **Sphere** dialog includes the following options:

Name lets you specify a unique identifier for the object. A descriptive name makes it easier to find a particular object in the Attribute Manager.

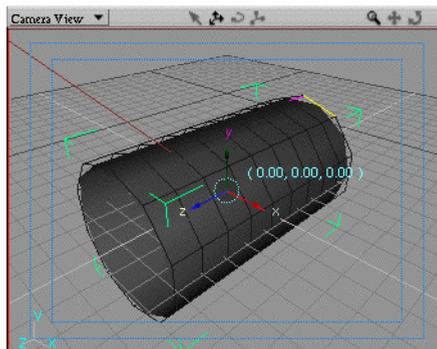
U_Steps lets you specify the number of sections in the sphere.

V_Steps lets you specify the number of rows of vertices from the top to the bottom of the sphere. The poles of the sphere count as one row each. The football above has **V_Steps** set to 40.

Note: A sphere must have at least 3 **U_Steps** and 3 **V_Steps**.

Radius lets you specify the distance between the center of the sphere and its surface in the **V** direction.

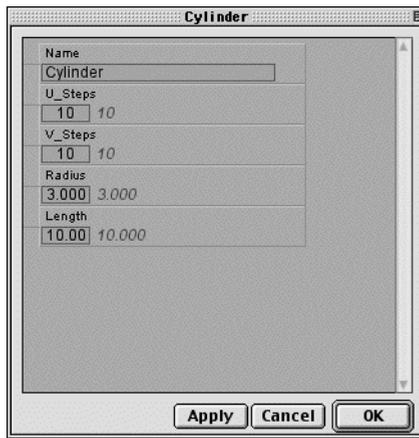
Cylinder



To create a cylinder...

1. Select **Cylinder** from the **NURBS** menu.
2. When the **Cylinder** dialog opens, you can use the default values or change them.

Figure 5.8: Cylinder Dialog



3. Click the OK button to create the cylinder.

Note: You can use the Attribute Manager to change the cylinder's attribute values later.

Cylinder Dialog Options. The Cylinder dialog includes the following options:

Name lets you specify a unique identifier for the object. A descriptive name makes it easier to find a particular object in the Attribute Manager.

U_Steps lets you specify the number of sections in the cylinder. For instance, you can create a rounded prism tube shape by setting **U_Steps** to 3. The rendered tube will have a much more rounded look.

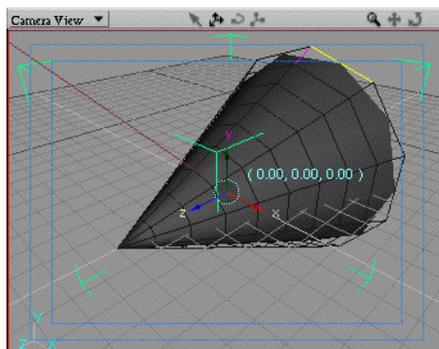
V_Steps lets you specify the number of rows of vertices from the start to the end of the cylinder. The start and end of the cylinder count as one row each.

Note: A cylinder must have at least 3 **U_Steps** and 3 **V_Steps** for it to be visible.

Radius lets you specify the distance between the center axis of the cylinder and its hull in the **U** direction.

Length lets you specify the distance between the start and end of the cylinder.

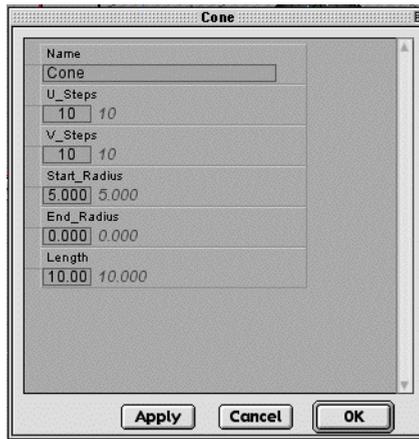
Cone



To create a cone...

1. Select **Cone** from the **NURBS** menu or click the **Cone** icon in the **Favorites Bar**.
2. When the **Cone** dialog opens, you can use the default values or change them.

Figure 5.9: Cone Dialog



3. Click the OK button to create the cone.

Note: You can use the Attribute Manager to change the cone's attribute values later.

Cone Dialog Options. The Cone dialog includes the following options:

Name lets you specify a unique identifier for the object. A descriptive name makes it easier to find a particular object in the Attribute Manager.

U_Steps lets you specify the number of sections in the cone. The effect of changing this value is most noticeable when **Periodic U** is turned off.

Figure 5.10: Cone with U_Steps=4 and Periodic U Turned Off

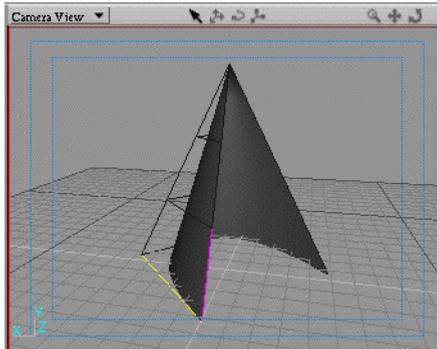
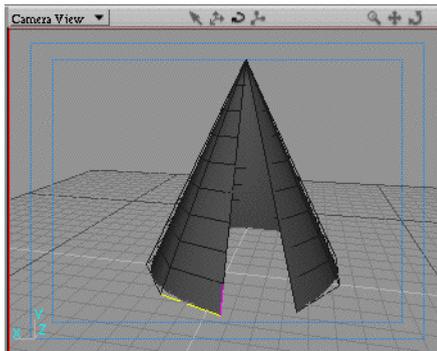


Figure 5.11: Cone with U_Steps=10 and Periodic U Turned Off



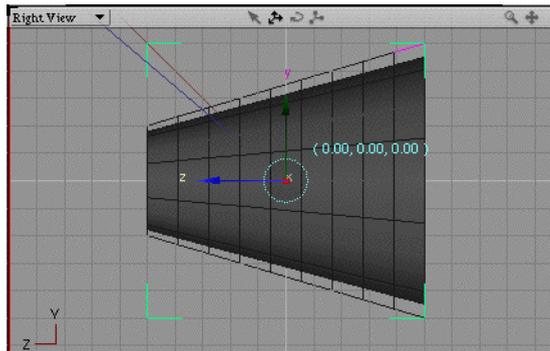
V_Steps lets you specify the number of rows of vertices from the start to the end of the cone. The start and end of the cone count as one row each.

Note: A cone must have at least 3 **U_Steps** and 3 **V_Steps**.

Start_Radius lets you specify the starting width of the cone.

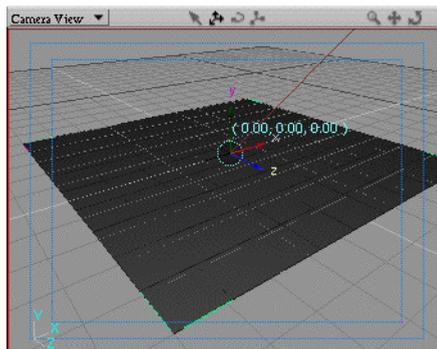
End_Radius lets you specify the ending width of the cone.

Figure 5.12: Cone with Start_Radius of 5 and End_Radius of 2



Length lets you specify the distance between the start and end of the cone.

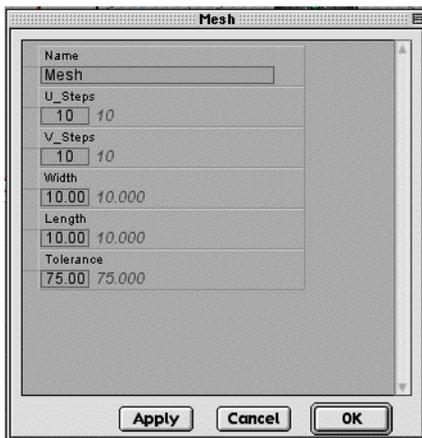
Mesh



To create a mesh...

1. Select **Mesh** from the **NURBS** menu or click the **Mesh** icon in the **Favorites Bar**.
2. When the **Mesh** dialog opens, you can use the default values or change them.

Figure 5.13: Mesh Dialog



3. Click the OK button to create the mesh.

Note: You can use the Attribute Manager to change the mesh's attribute values later

Mesh Dialog Options. The **Mesh** dialog includes the following options:

Name lets you specify a unique identifier for the object. A descriptive name makes it easier to find a particular object in the Attribute Manager.

U_Steps lets you specify the number of rows of vertices on the width of the mesh. The edges of the mesh count as one row each.

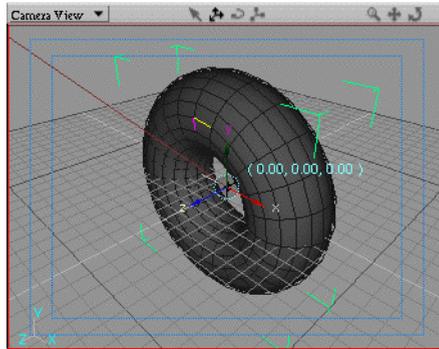
V_Steps lets you specify the number of rows of vertices on the length the end of the mesh. The start and end of the mesh count as one row each.

Note: A mesh must have at least 3 **U_Steps** and 3 **V_Steps** for it to be visible.

Width lets you specify the distance along the **U** axis.

Length lets you specify the distance along the V axis.

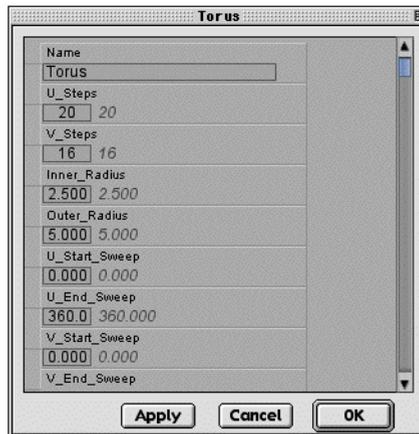
Torus



To create a torus...

1. Select Torus from the **NURBS** menu or click the Torus icon in the Favorites Bar.
2. When the Torus dialog opens, you can use the default values or change them.

Figure 5.14: Torus Dialog



3. Click the OK button to create the torus.

Note: You can use the Attribute Manager to change the torus's attribute values later.

Torus Dialog Options. The Torus dialog includes the following options:

Name lets you specify a unique identifier for the object. A descriptive name makes it easier to find a particular object in the Attribute Manager.

U_Steps lets you specify the number of sections in the torus. The effect of changing this value is most noticeable when **Periodic U** is turned off.

V_Steps lets you specify the number of rows of vertices around the surface of the torus.

Note: The cross-section is revealed by un-checking **Periodic U** in the **NURBS** attributes of the torus.

Note: A torus must have at least 4 **U_Steps** and 4 **V_Steps**.

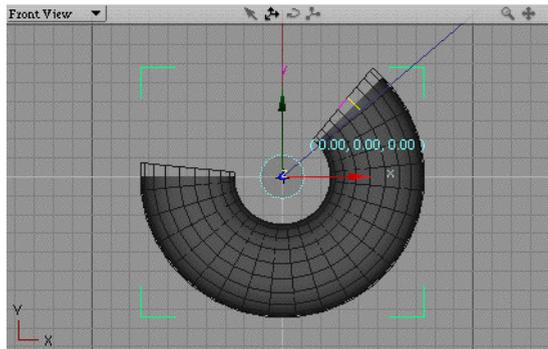
Inner_Radius lets you determine the thickness of the torus by specifying the radius of its cross-section.

Outer_Radius lets you specify the distance between the center of the hole in the torus and the edge of the hull of the torus farthest from that center point.

Note: You can create a round cushion shape by making the **Inner_Radius** bigger than the **Outer_Radius**.

U_Start_Sweep and **U_End_Sweep** let you control how much of the torus to create in the **U** direction. By specifying different starting and ending positions, in degrees clockwise, you can create an arc of a torus. For instance, you could create a partly-eaten donut by setting **U_Start_Sweep** to 45 and **U_End_Sweep** to 270.

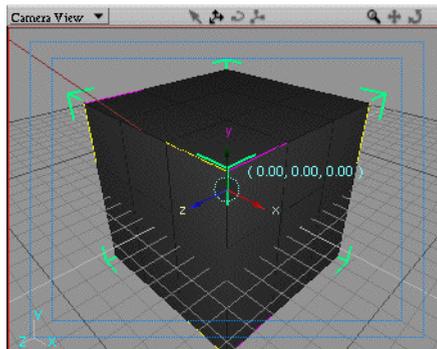
Figure 5.15: Partly-eaten Donut



`V_Start_Sweep` and `V_End_Sweep` let you control how much of the torus to create in the `V` direction. By specifying the starting and ending positions, in degrees clockwise, you can create a torus section.

Cube

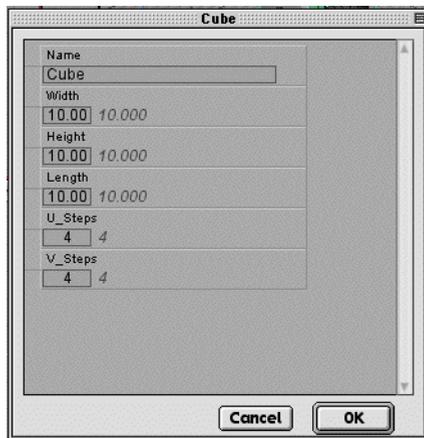
A NURBS cube is actually 6 NURBS meshes all linked to a null. If you want an object with a single surface you can use a **Cube Solid** object instead of a NURBS cube. In some cases you can instead use the **Cubeoid** script on the **Scripts** menu to create a cube-like NURBS object that has a single surface and slightly rounded corners.



To create a cube...

1. Select **Cube** from the **NURBS** menu or click the **Cube** icon in the **Favorites Bar**.
2. When the **Cube** dialog opens, you can use the default values or change them.

Figure 5.16: Cube Dialog



3. Click the **OK** button to create the cube.

Note: You can use the **Attribute Manager** to change the cube's attribute values later.

Cube Dialog Options. The **Cube** dialog includes the following options:

Name lets you specify a unique identifier for the object. A descriptive name makes it easier to find a particular object in the **Attribute Manager**.

Width lets you specify the distance along the **U** axis.

Height lets you specify the height of the cube.

Length lets you specify the distance along the **V** axis.

U_Steps lets you specify the number of rows of vertices on the width of the cube. The edges of the cube count as one row each.

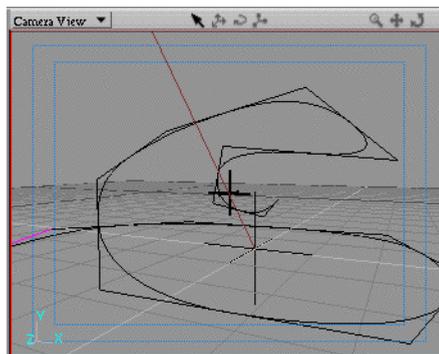
V_Steps lets you specify the number of rows of vertices on the length the end of the cube. The start and end of the cube count as one row each.

Note: A cube must have at least 3 **U_Steps** and 3 **V_Steps** for it to be visible.

CURVES

Freehand

Figure 5.17: Freehand Curve

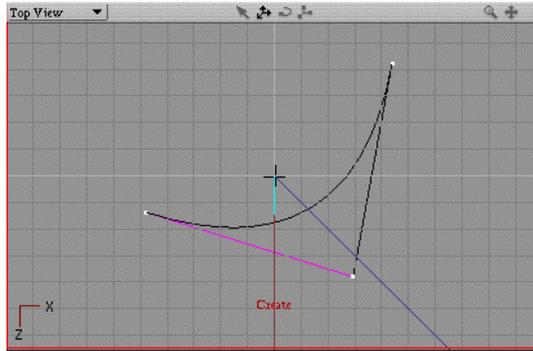


To create a freehand curve...

1. Click the **Freehand** tool on the **Favorites Bar**. The word **Create** appears in red at the bottom of each view pane to indicate the **Freehand** tool is active.
2. Click in the view pane where you wish to start drawing. You can start in any of the views that are visible. A white dot appears to indicate the first control point of the curve.
3. Click to create a second control point. A black line is drawn between the two points.

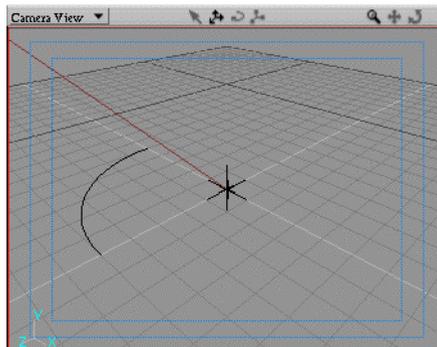
4. Click to create a third control point. This determines the first segment of the freehand curve.

Figure 5.18: Three-point Freehand Curve



5. Continue adding control points until you finish the curve. You can click in any view pane to add a control point there.
6. Hit the `esc` key to end the curve.

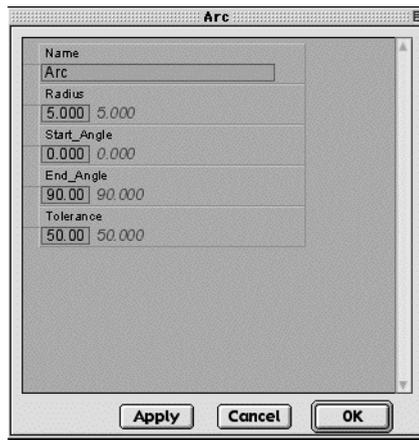
Arc



To create an arc...

1. Select **Arc** from the **Curves** menu.
2. When the **Arc** dialog opens, you can use the default values or change them.

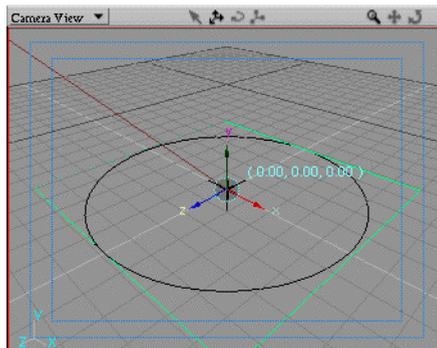
Figure 5.19: Arc Dialog



3. Click the OK button to create the arc.

Note: You can use the Attribute Manager to change the arc's attribute values later.

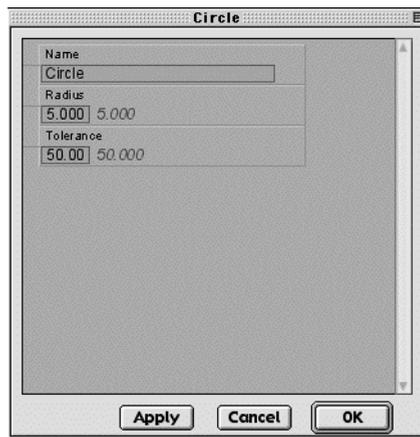
Circle



To create a circle...

1. Select Circle from the Curves menu.
2. When the Circle dialog opens, you can use the default values or change them.

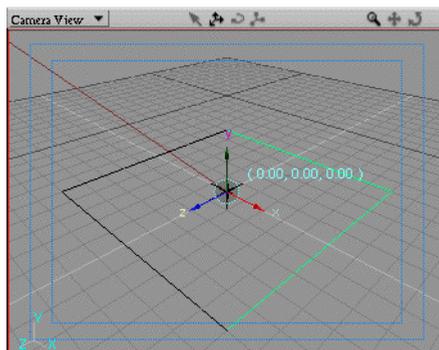
Figure 5.20: Circle Dialog



3. Click the OK button to create the circle.

Note: You can use the Attribute Manager to change the circle's attribute values later

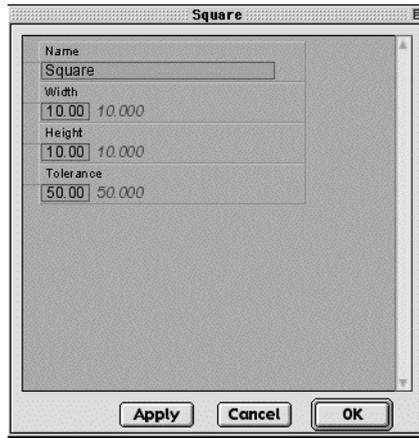
Square



To create a square...

1. Select Square from the Curves menu.
2. When the Square dialog opens, you can use the default values or change them.

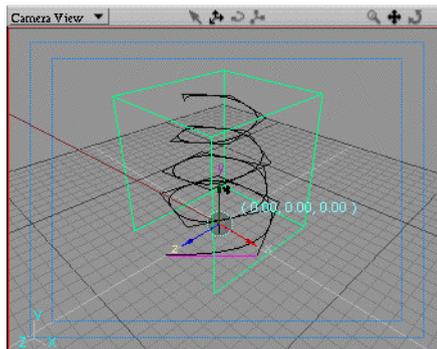
Figure 5.2 I : Square Dialog



3. Click the OK button to create the square.

Note: You can use the Attribute Manager to change the square's attribute values later

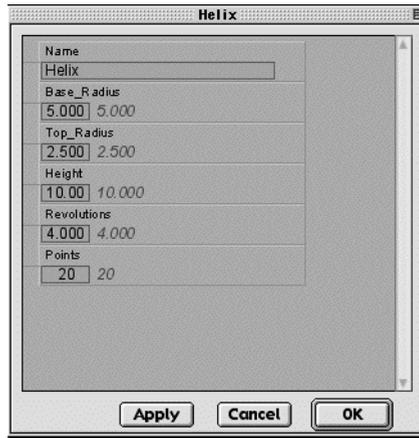
Helix



To create a helix...

1. Select **Helix** from the **Curves** menu.
2. When the **Helix** dialog opens, you can use the default values or change them.

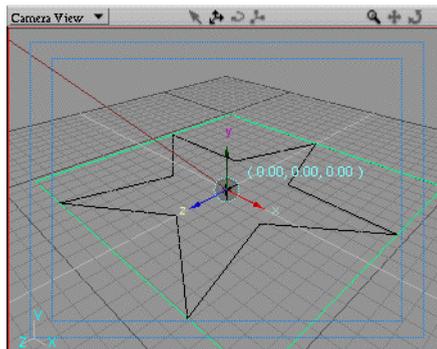
Figure 5.22: Helix Dialog



3. Click the OK button to create the helix.

Note: You can use the Attribute Manager to change the helix's attribute values later

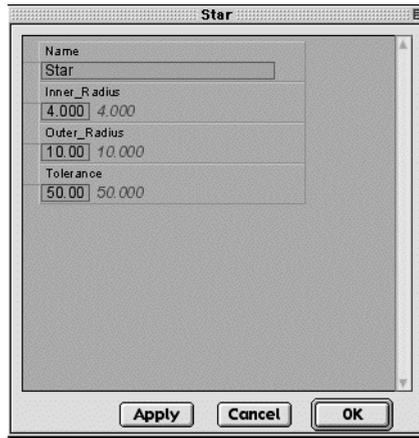
Star



To create a star...

1. Select **Star** from the **Curves** menu.
2. When the **Star** dialog opens, you can use the default values or change them.

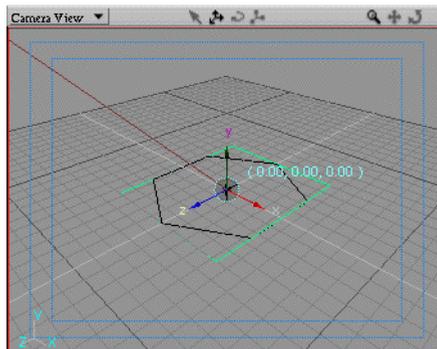
Figure 5.23: Star Dialog



3. Click the OK button to create the star.

Note: You can use the Attribute Manager to change the star's attribute values later

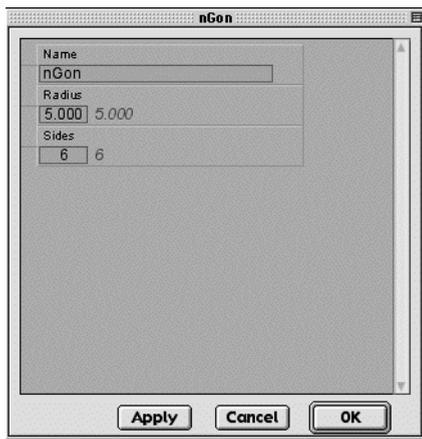
nGon



To create a polygon (ngon)...

1. Select nGon from the *Curves* menu.
2. When the nGon dialog opens, you can use the default values or change them.

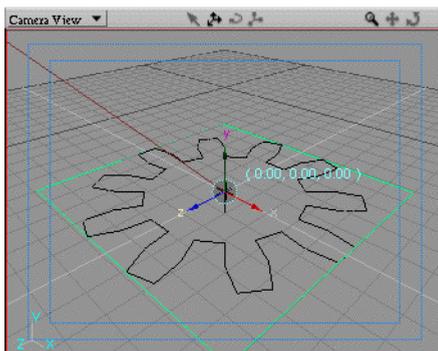
Figure 5.24: nGon Dialog



3. Click the OK button to create the polygon.

Note: You can use the Attribute Manager to change the polygon's attribute values later

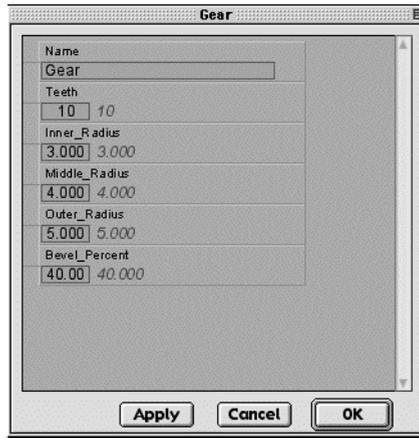
Gear



To create a gear...

1. Select **Gear** from the **Curves** menu.
2. When the **Gear** dialog opens, you can use the default values or change them.

Figure 5.25: Gear Dialog



3. Click the OK button to create the gear.

Note: You can use the Attribute Manager to change the gear's attribute values later.

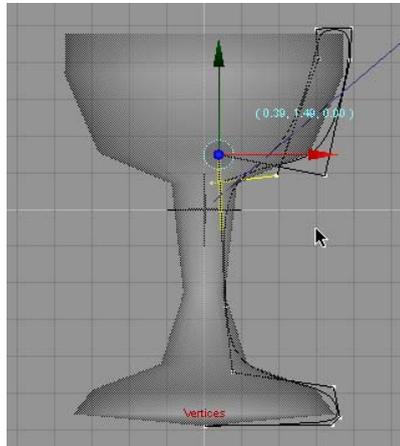
Chapter 6. Constructed NURBS Objects

A constructed NURBS objects can be created from a base NURBS curve or editable freehand spline. When you construct an object, the curves you used in the construction remain in your scene.

To change the shape of a constructed object...

Edit the curve(s) used to construct the object. The constructed shape will dynamically change in response to your edits.

Editing a Lathe



(Click the picture to view the animation)

To use Edit NURBS tools on a constructed object...

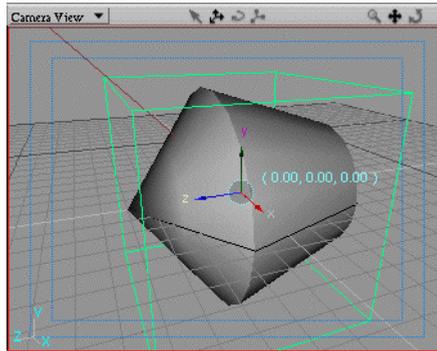
1. Select the constructed object.
2. Choose **Make Editable** from the **Edit NURBS** menu. This will create a new editable NURBS object that has the same shape as your constructed object. The constructed object

will be kept for future use, but hidden so it does not clutter up your view.

Lathe

A lathed object is a surface created by spinning a base curve around an axis. Creating a lathed object from a closed curve produces a solid shape.

Figure 6.1: Pentagon Lathed Around an Axis

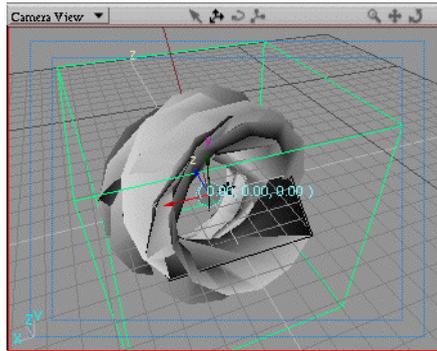


To create a lathed object..

1. Create a NURBS curve or editable spline.
2. Select the base curve.
3. Select the view corresponding to the axis you want to lathe the curve around:
 - the Top view for the Z axis
 - the Front view for the X axis
 - the Right view for the Y axis
4. Select **Lathe** from the **NURBS** menu or click the **Lathe** icon in the **Favorites Bar**.

Note: Unexpected shapes will result if you select the **Camera** view and then lathe the curve. The shape below was created from the same pentagon as the shape above.

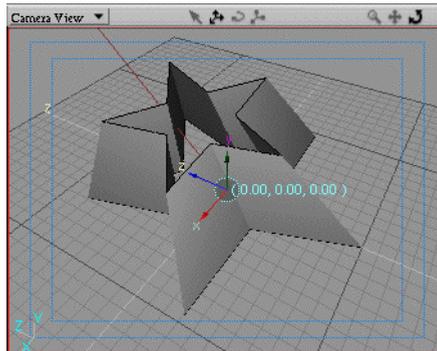
Figure 6.2: Pentagon Lathed from the Camera View



Loft

A lofted object is a shape determined by contours on its surface. Each contour curve *must* have the same number of points!

Figure 6.3: Lofted Star



To create a lofted object...

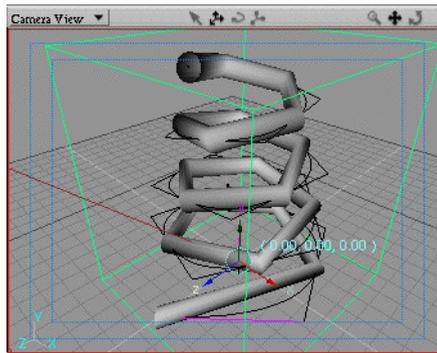
1. Create a NURBS curve (by selecting one of the options from the **Curves** menu) or editable spline (by selecting a NURBS curve and choosing **Make Editable** from the **Edit NURBS** menu).
2. Duplicate the first curve and move the duplicate to produce the next contour curve. If you want, you can edit the next contour curve, just do not add or delete any vertices.

3. Continue duplicating and moving curves to produce as many contours as you want.
4. With no objects selected, select **Loft** from the **NURBS** menu or click the **Loft** icon in the **Favorites Bar**.
5. Select the first curve.
6. Select the next contour curve.
7. Continue selecting contour curves in order until all of the contour curves have been selected.
8. Use the **Escape** key to indicate when you are done selecting curves.

Sweep

A swept object is a surface created by moving a curve along a path.

Figure 6.4: Circle Swept along Helix



To create a swept object...

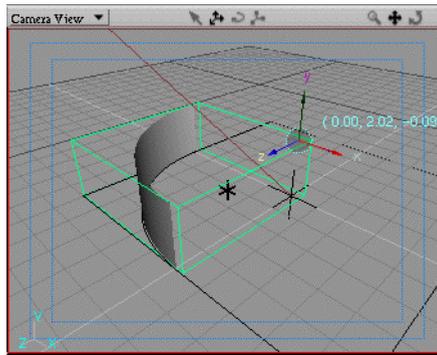
1. Create a NURBS curve or editable spline for the base curve, which will form the object's cross-section.
2. Create a NURBS curve or editable spline for the path.
3. With no objects selected, select **Sweep** from the **NURBS** menu or click the **Sweep** icon in the **Favorites Bar**.
4. Select the base curve.

Extrude

5. Select the path curve.

An extruded object is the surface created by moving a base NURBS curve along a straight path.

Figure 6.5: Extruded Arc



To create an extruded object...

1. Create a NURBS curve or editable spline.
2. Select the base curve.
3. Select **Extrude** from the **NURBS** menu or click the **Extrude** icon in the **Favorites Bar**.

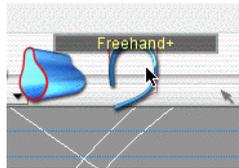
Chapter 7. Tutorial: Flying the Flag

In this simple tutorial you will create a flag by drawing a curve, extruding the curve then putting a bitmap image on the extruded surface.

Modeling the Flag

Start by drawing the curve.

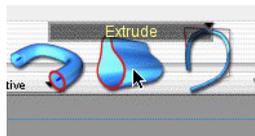
1. Select **Freehand**



2. Draw a curvy line by clicking in the **Top View** for each vertex on the line. When you are finished, press **Esc**.

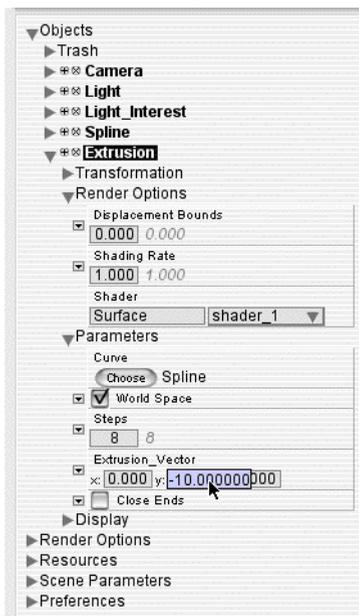


3. Select **Extrude**

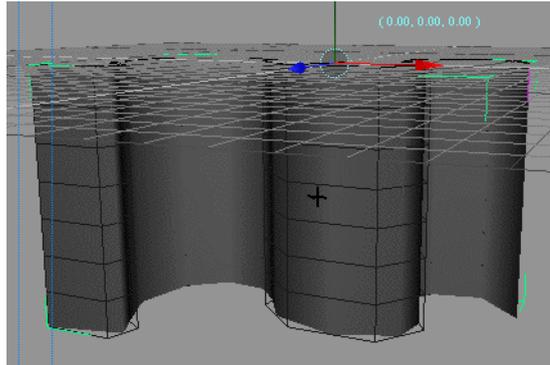


4. In the Attribute Manager (the list in the lower right-hand portion of the window), click the arrow next to **Objects** to

expand the object list. **Extrusion** should be selected. Click the triangle next to that to display the attributes, and then **Parameters** to display the attributes that describe the spline. Change **y** in the **Extrusion Vector** to **-10** by clicking in the box with the value, typing **-10**, and hitting Return. The **x** and **z** values should be **0**. Notice that the **Extrusion** object changes on screen as you make the changes in the Attribute Manager.



You now have an extruded surface in the shape of a flag.



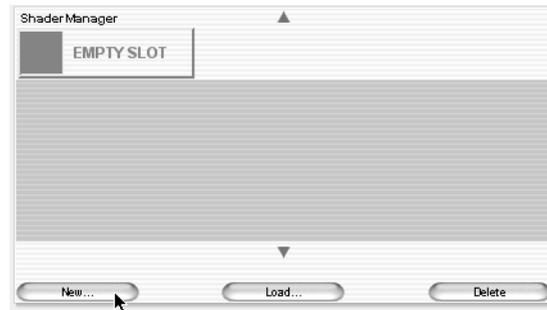
Shading the Flag

You are now going to incorporate a bitmap into a shader to color the flag.

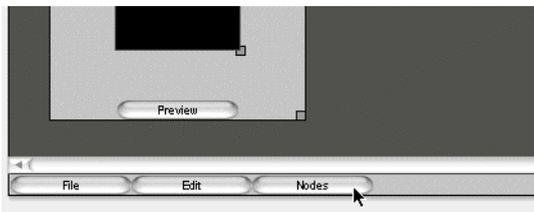
1. Switch to the **Shader** layout by clicking the **Shader Layout** icon (which looks like a paintbrush).



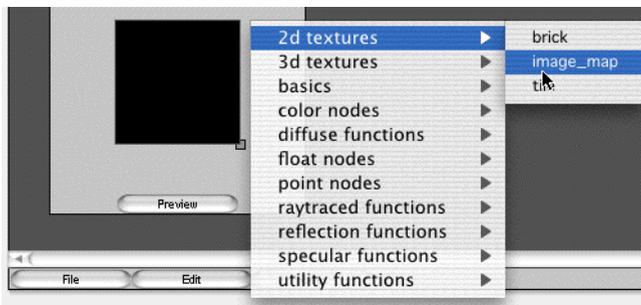
2. Click **New** in the **Shader Manager**. This creates a new default shader. The shader contains a single little window titled **Object_#**. The little windows in a shader are called nodes. They are the building blocks of shaders. You construct a shader by creating nodes and connecting the output of one node to the inputs of other nodes.



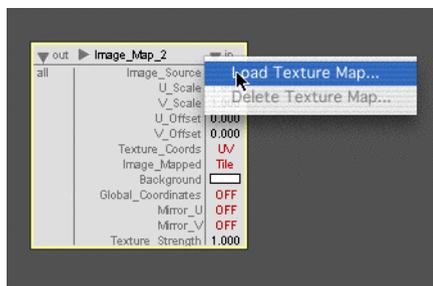
3. Click on the **Nodes** button at the bottom of the **Shader Maker** panel. This opens the **Shader Node** menu.



4. Select **2d textures > image_map** from the **Shader Node** menu. This creates a new **Image Map** node.

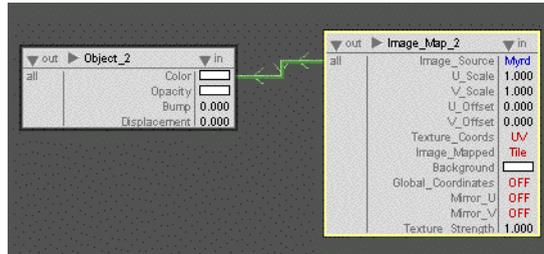


5. Click on the **Image_Source** field in the **Image Map** node and select **Load Texture Map**.

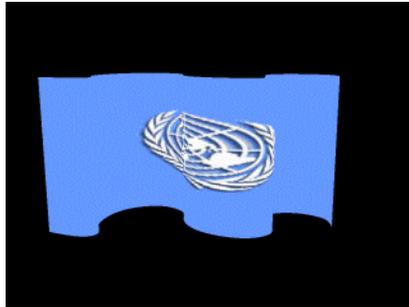


6. With the resulting **File Open** dialog choose an image map. PiXELS 3D supports a wide variety of image map formats including jpg, tiff, and pct. This tutorial uses an image of the United Nations flag, but you can choose any image you want.

7. Drag **all** from under **out** on the **Image Map** node to **Color** under **in** on the **Object** node. This created a line (green in the illustration below) from the **Image Map** node to the **Object** node.



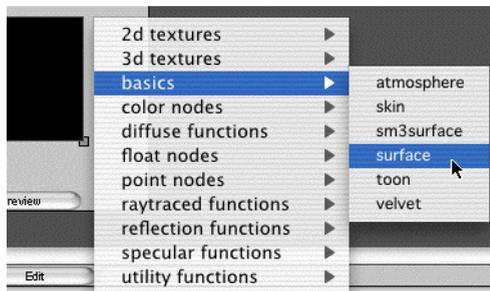
8. Now select **Render** from the **File** menu. The bitmap shows up nicely on the rendered loft, but it has no highlighting or shadows, and the image is backwards.



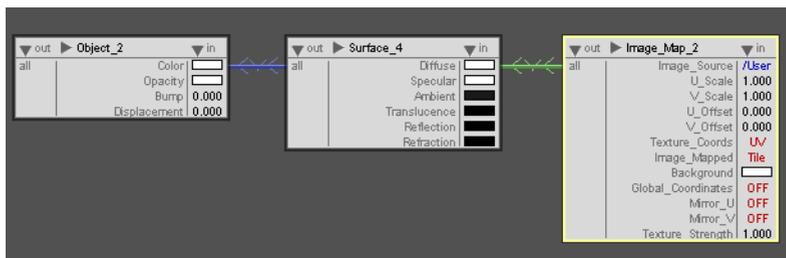
Highlighting and Shadows

Any time you want highlighting and shadows you need to incorporate at least one surface node into a shader. The surface nodes are: **Surface**, **SM3Surface**, **Skin**, **Toon**, **Velvet**, **Clay**, and **Diffuse**.

1. Choose **basics > surface** from the **Shader Nodes** menu to create a new **Surface** node.

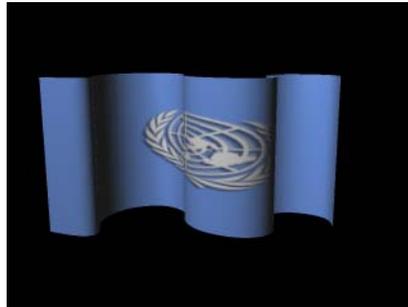


2. Drag **all** from under **out** on the **Surface** node to **Color** under **in** on the **Object** node. This creates a line (blue in the illustration below) from the **Surface** node to the **Object** node. Drag **all** from under **out** on the **Image Map** node to **Diffuse** on the **Surface** node. This creates a line (green in the illustration below) from the **Image Map** node to the **Surface** node.



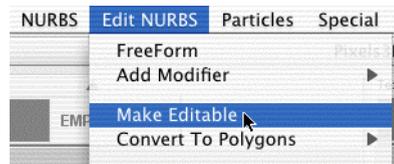
3. Now select **Render** from the **File** menu. The resulting image now has highlights and shadows; however, the image is still backwards. (You can tell the image is backwards because South America is on the right, or eastern hemi-

sphere rather than the left, or western hemisphere where it belongs.)

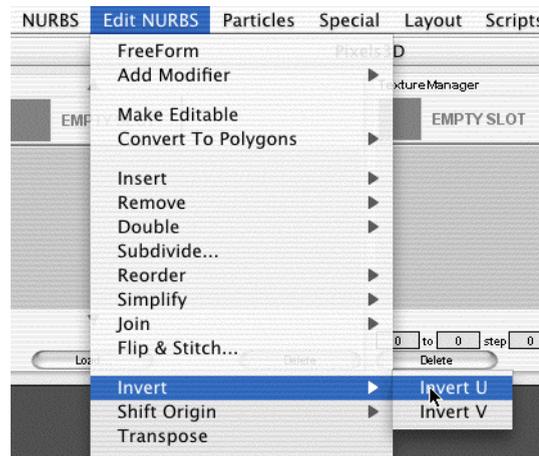


Orientation

1. To flip the image, you need to flip the geometry of the tag. To do this, you first need to make the extrusion into an editable NURBS object. Select **Make Editable** from the **Edit NURBS** menu. This creates a new NURBS object with the same geometry as the extrusion, then hides the extrusion object. The new NURBS object will be called **Extrusion_1** by default.



2. To flip the geometry, select the new NURBS object and choose **Invert U** from the **Edit NURBS** menu.



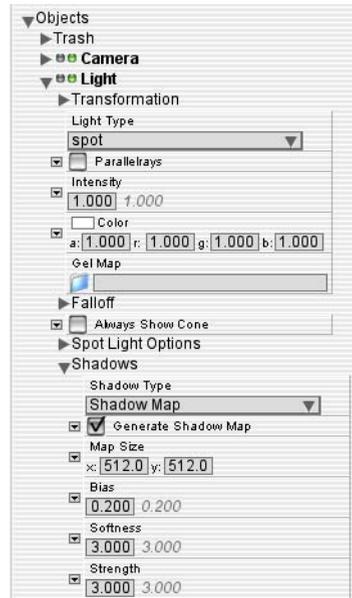
3. Click on **Shader_1** to assign the image map shader you were working on to the new NURBS object.
4. Select **Render** from the **File** menu. The image map is now in the correct orientation.



Casting Shadows

1. To get shadows cast across the flag, you need to tell the light source that it is supposed to cast shadows. Click the triangle next to **Light** in the Attribute Manager, and then the triangle next to **Shadows**. Select **Shadow Map** as the

Shadow Type. Under that check the Generate Shadow Map box

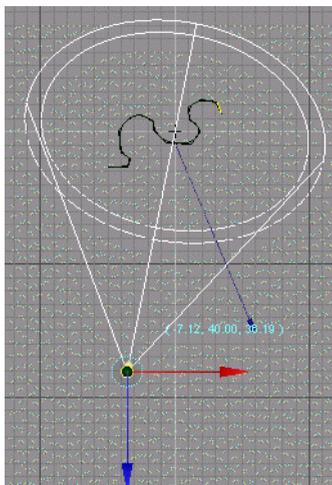


Note: If you want to use raytraced shadows rather than a shadow map, select **Raytraced** as the **Shadow Type**. Raytraced shadows are more accurate than shadow maps, but are often slower to render.

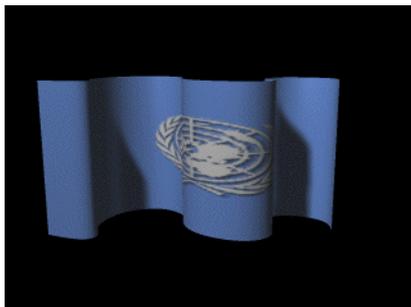
2. To enhance the shadows on the bag, you will want to move the light. In the **Top View**, zoom out until you can see

Specularity

the light, which should still be selected. Move the light to the left



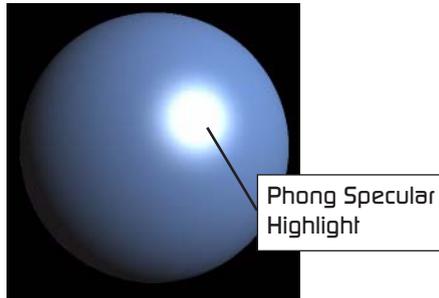
3. When you render again, the bag has shadows across it. This image is good enough to use; however, the material looks like plastic. A bag should be something that looks a little more like cloth. To do this, you can change the specular highlights.



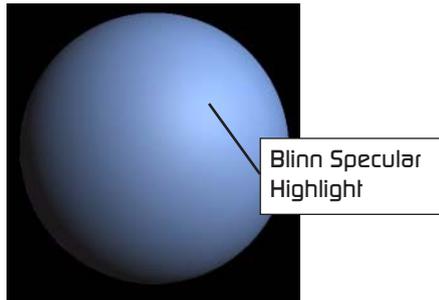
Specularity

Specularity is the way bright spots appear on a surface in response to lighting. A surface without any specular highlights is a 100% diffuse non-glossy surface. By default, sur-

faces use Phong specularity, which makes the surface look like plastic.

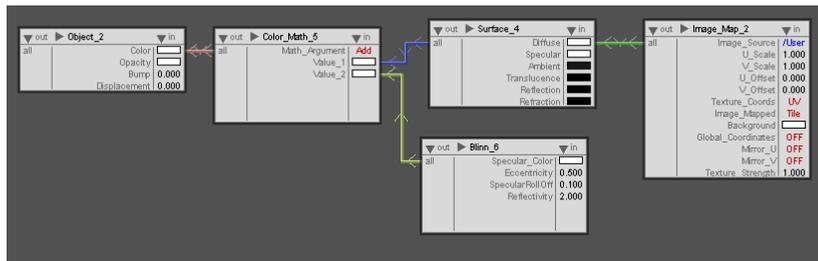


Blinn specularity has a more matte or semi-gloss appearance.



1. Add a **Blinn** specular node: from the **Nodes** menu, select **specular functions** and then **blinn**.
2. To combine the **Blinn** node with the **lag** surface, add a **Color Math** node: from the **Nodes** menu, select **color nodes** and then **color math**.
3. Connect the **Blinn** node to one input of the **Color Math** node and connect the **Surface** node to the other input of the **Color Math** node.
4. Connect the **Color Math** node output to **Color** on the **Object** node and make sure the **Color Math** node **Math Argument** is **Add**.

5. In the Blinn node change the Reflectivity value to 2.



6. When you render again the object looks less glossy, more like paper or a woven material than a sheet of plastic.



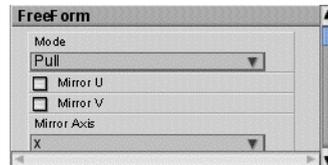
Chapter 8. NURBS Modeling

FreeForm

FreeForm tools allow you to manipulate your objects a little like you might model clay. When you push your thumb into a ball of clay or putty, the point where you push in influences nearby points on the ball's surface, so you make a big dent rather than a sharply defined hole. Likewise, with **FreeForm** tools, when you move a control vertex, it influences nearby control vertices, so you dent and stretch the surface rather than creating sharp bumps and valleys.

To use a **FreeForm** tool...

1. Choose **Edit NURBS>FreeForm**. This switches you to **Vertices** display mode and switches you to a **FreeForm** tool.
2. Make any adjustments to the tool in the **Tool Attributes** panel (the panel will be titled **FreeForm**).



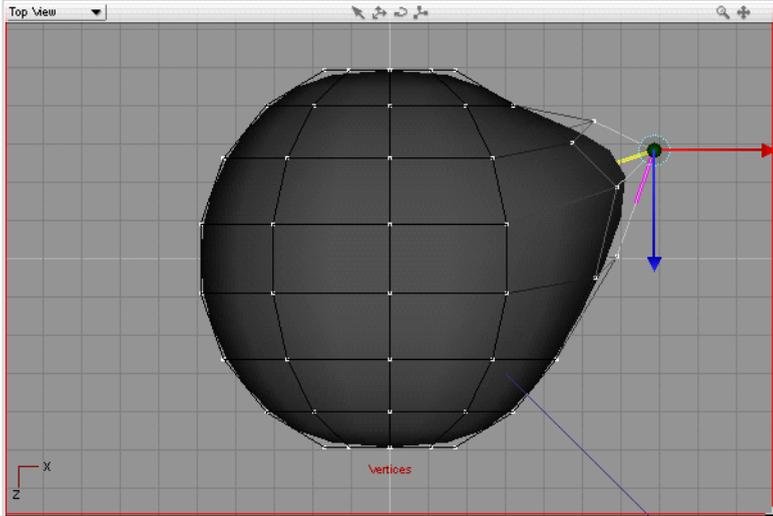
3. Select and drag vertices to edit your object.

Mode

There are two very different **FreeForm** tools which you can choose from the **Mode** drop-down on the **FreeForm Tool Attributes** panel.

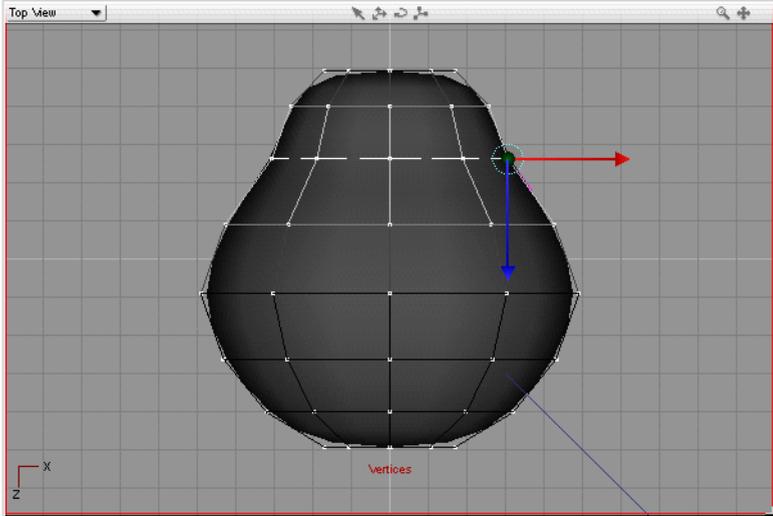
Pull. With this tool you can make bumps and dents by dragging control vertices. Notice that the lines closest to the selected vertex are lighter than the ones further away. The brightness of the lines is a visual indication of how much influence moving the selected vertex will have on the other vertices.

Figure 8.1: Pulled Vertex



Pinch. With this tool you can expand or pinch an object's waist (u dimension) by clicking on a vertex and dragging the mouse up or down.

Figure 8.2: Pinched Object

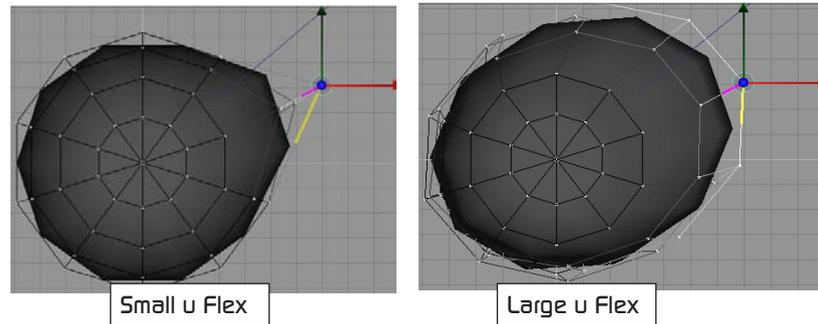


Flex

Flex controls how much moving one vertex affects nearby vertices.

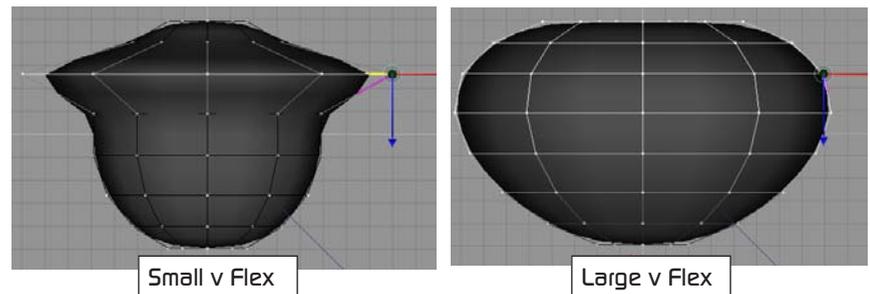
u Flex. This only affects the **Pull** tool, not the **Pinch** tool. A larger **u Flex** makes a wider dip or hill in the **u** dimension when you **Pull** a vertex. Notice how the brightness of the lines changes as the influence on the vertices changes.

Figure 8.3: Small and Large u Flex



v Flex. This affects both **FreeForm** tools. A larger **v Flex** makes a wider dip or hill in the **v** dimension when you **Pull** or **Pinch** a vertex.

Figure 8.4: Small and Large v Flex

**Shape Curve**

The **Shape Curve** controls how **u** and **v** flex diminish with distance.

Figure 8.5: Smooth Shape Curve

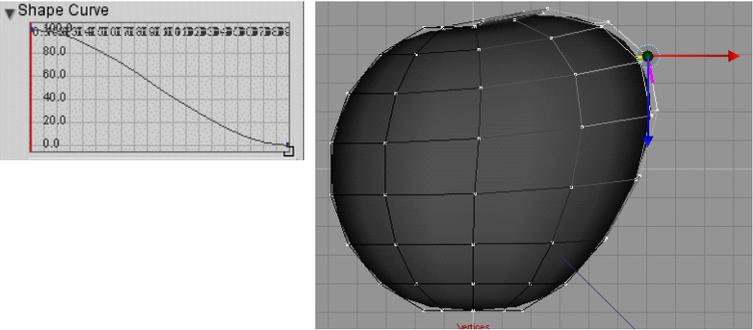


Figure 8.6: Bowed Shape Curve

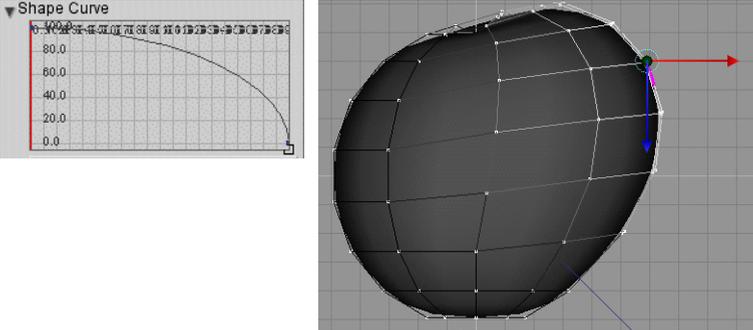
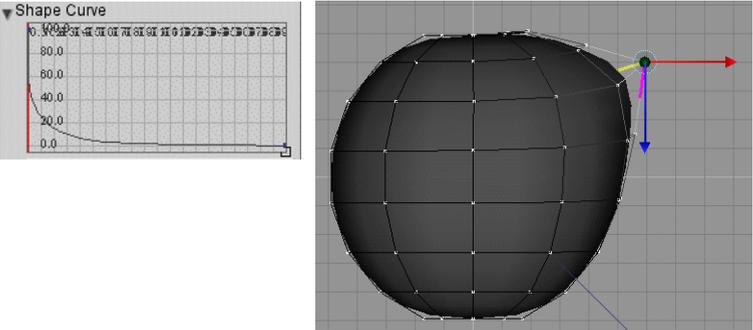


Figure 8.7: Cupped Shape Curve



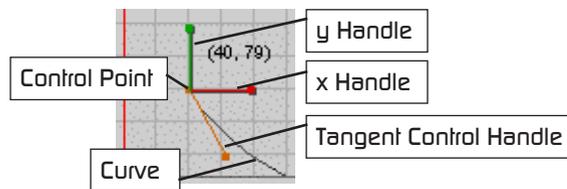
To edit the shape curve...

1. Click on one of the control points. (There are always control points at either end of the curve. You may have also added a control point to the curve.) This reveals the orange tangent control handle and the red and green x and y handles.
2. Click and drag the end of the orange tangent control handle. The curve remains tangent to the handle.

or

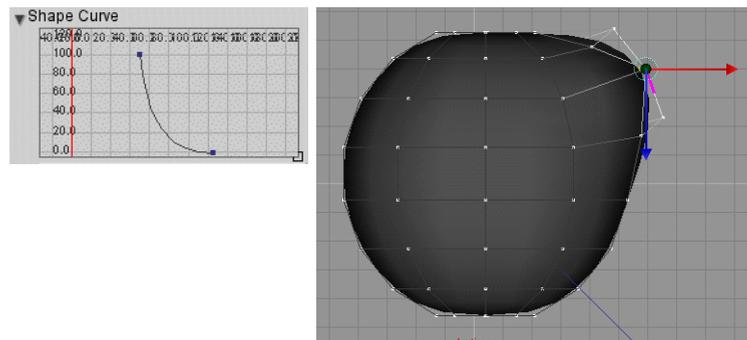
Click and drag the control point or the x or y handle to move the control point.

Figure 8.8: Shape Curve Controls



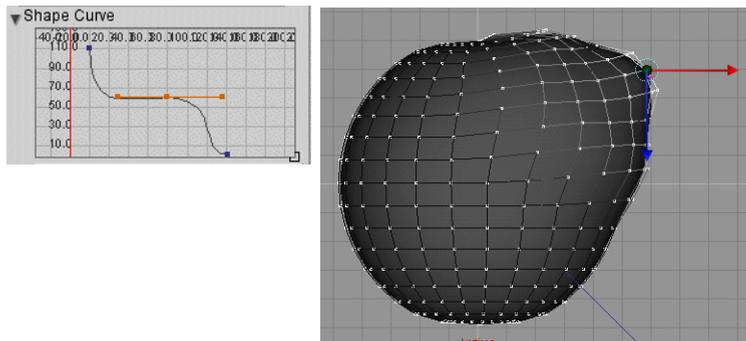
By moving the end points you can make the shape curve more cliff-like.

Figure 8.9: Cliff-like Curve



To create a new control point,
Cmd-Click on the curve.

Figure 8.10: Control Point Added



To delete a control point,
Cmd-Click on the control point.

Modifiers

Modifiers are collections of attributes. Adding a modifier allows you to manipulate an object by making changes to the modifier in the **Attribute Manager**.

To use a modifier...

1. Select the object you want to modify.
2. Choose the modifier you want to use from the **Edit NURBS>Add Modifier** menu.
3. In the **Attribute Manager** go to the **Modifiers** attribute of the object and adjust the modifier you added.

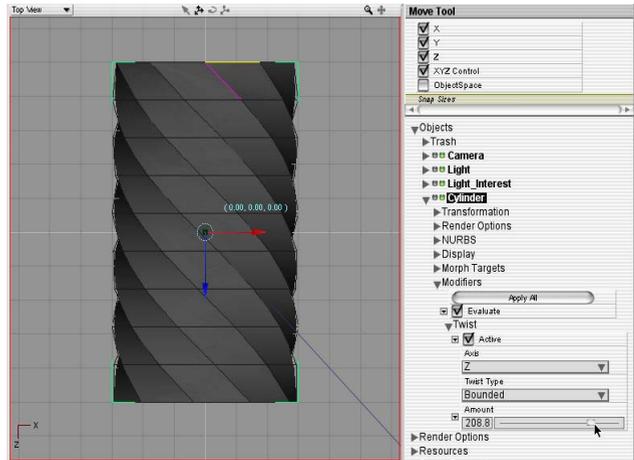
Note: All modifiers have an **Active** checkbox. To temporarily turn off the effect, uncheck **Active**.

The best way to understand what each attribute does is to try adjusting it.

Twist

The animation below shows how the Attribute Manager controls the amount of twist in a Twist Modifier.

Figure 8.11: Twist Modifier



(Click the picture to view the animation)

Taper

The image in Figure 8.13 was produced with the Taper attributes in Figure 8.12.

Figure 8.12: Taper Attributes

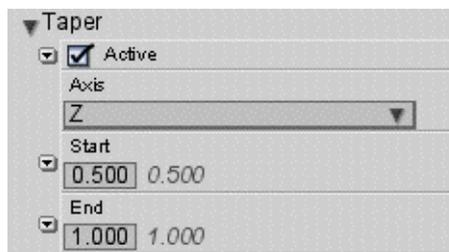
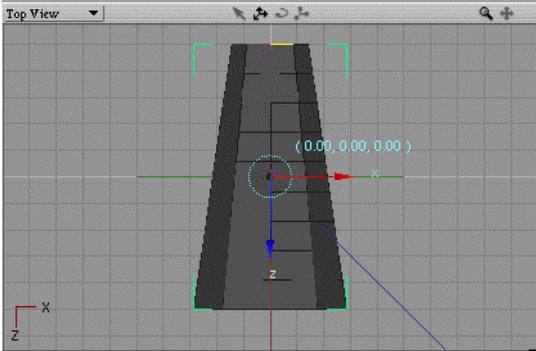


Figure 8.13: Tapered Object



Wave

The image in Figure 8.15 was produced with the Wave attributes in Figure 8.14.

Figure 8.14: Wave Attributes

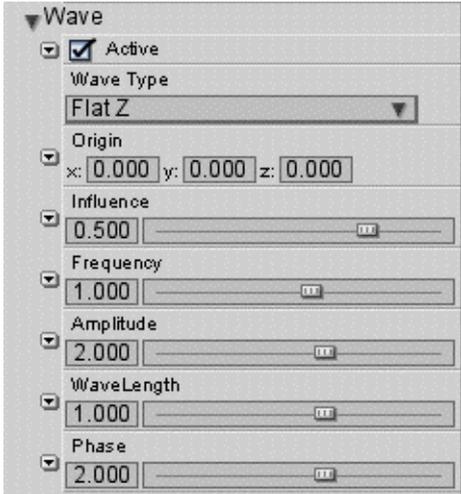
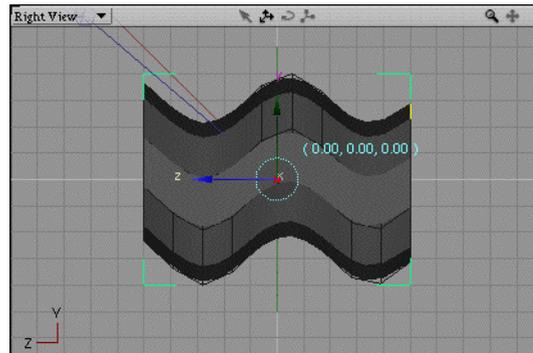


Figure 8.15: Wavy Object

**Disrupt**

The image in Figure 8.15 was produced with the Disrupt attributes in Figure 8.14.

Figure 8.16: Disrupt Attributes

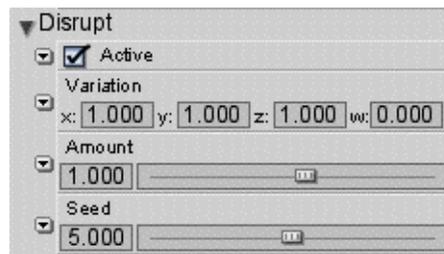
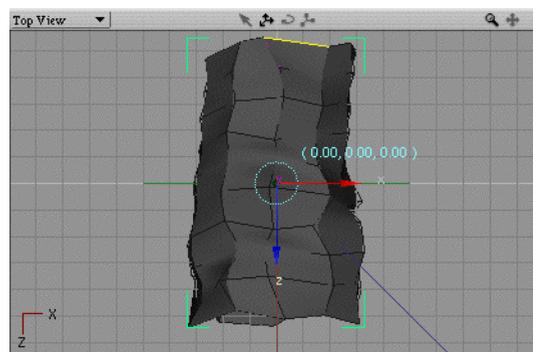


Figure 8.17: Disrupted Object



Modifiers

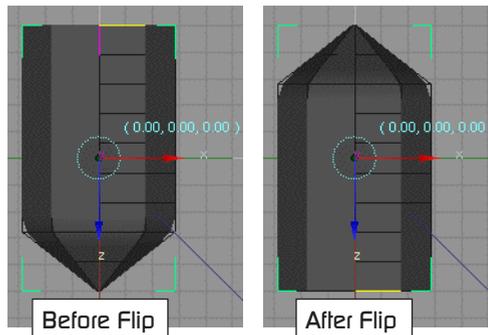
Flip

The image in Figure 8.19 shows an object before and after being flipped. Figure 8.18 shows the attributes used to flip the object.

Figure 8.18: Flip Attributes



Figure 8.19: Flipped Object



Shear

The image in Figure 8.21 was produced with the Shear attributes in Figure 8.20.

Figure 8.20: Shear Attributes

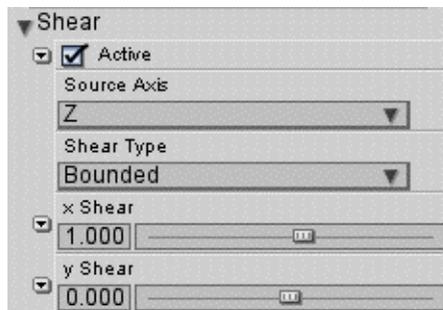
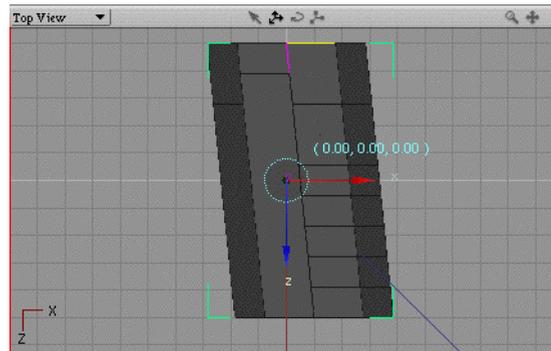


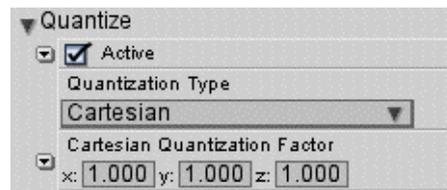
Figure 8.21: Sheared Object



Quantize

The image in Figure 8.21 was produced with the Quantize attributes in Figure 8.22.

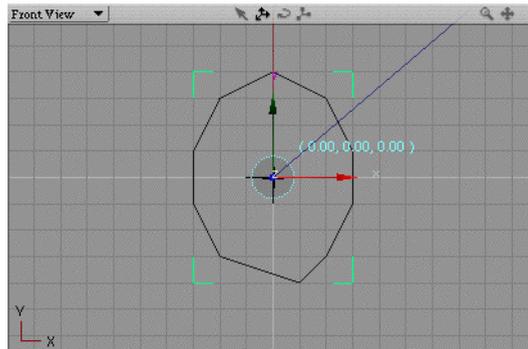
Figure 8.22: Quantize Attributes



In the figures the Quantization Type is Cartesian which means all vertices are snapped to the nearest x , y , z coordinate evenly divisible by the Cartesian Quantization Factor.

The alternative Quantization Type is Polar which means the vertices are snapped to points at a distance from the center which is evenly divisible by the Polar Quantization Factor.

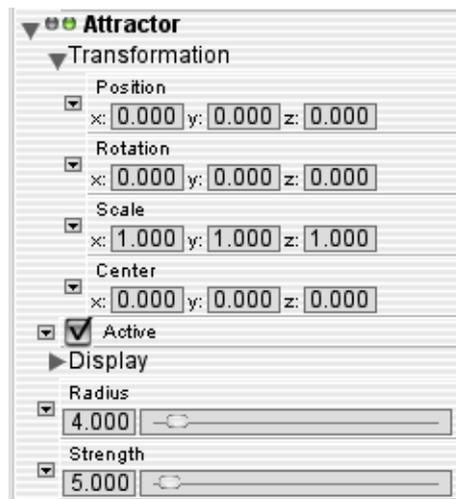
Figure 8.23: Quantized Vertices



Attractor

The Attractor modifier is useful for moving vertices in an animation. The attractor pulls vertices towards itself.

Figure 8.24: Attractor Attributes



Radius. The area of effect of the attractor is determined by its radius.

Strength. Controls how much the attractor affects the vertices.

Bulge

The **Bulge** modifier provides a useful shortcut for modeling. It can also be used for squash-and-stretch animation.

Figure 8.25: Bulge Attributes

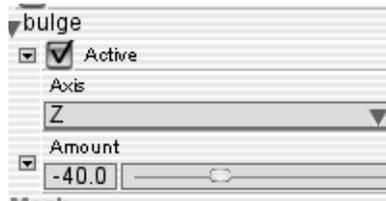
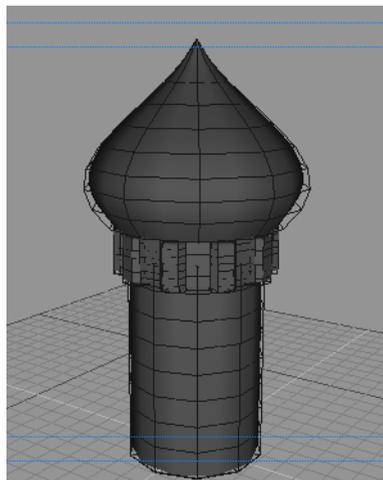
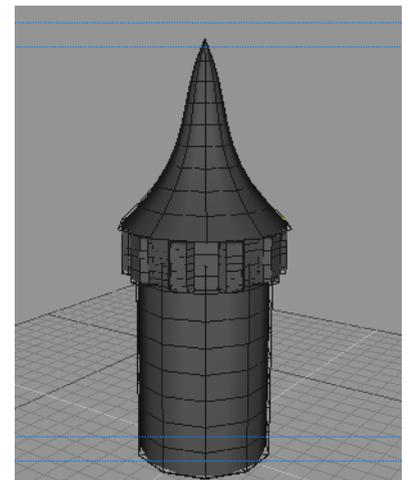


Figure 8.26: Bulge Applied to a Cone



Bulge Amount = 100



Bulge Amount = -40

Deformers

Deformers are covered in [CharacterAnimation Tools](#) on page 319.

Conversion

Make Editable. When you make a **Solid**, any kind of **Curve** except for a **Freehand Curve**, or construct a NURBS object like a **Lathe**, **Loft**, **Sweep**, or **Extrude**, the object is not editable. That is, there are no vertices on these objects that you can move or

otherwise edit. You can create an editable NURBS object from any of these objects by selecting **Make Editable**. PiXELS will hide the original object.

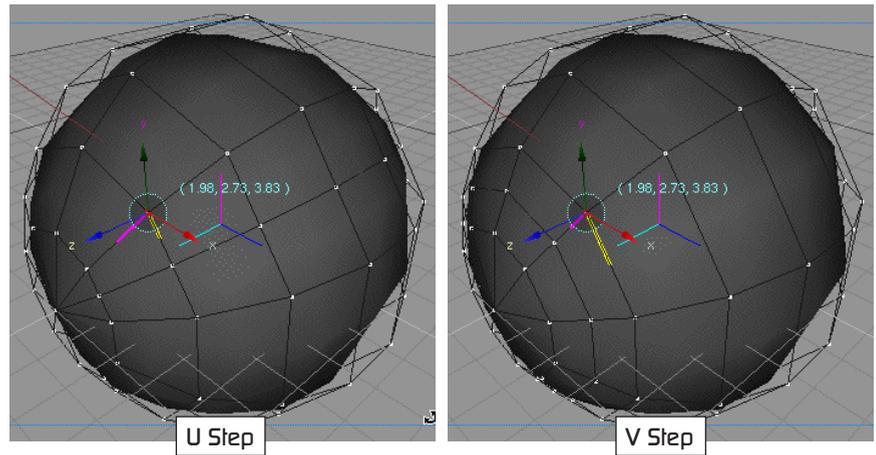
Convert to Polygons>Tessellate. Creates a mesh of triangles that forms a polygonal object from the selected object. You will have to hide or delete the original object if you do not want it to show up in your scene.

Convert to Polygons>Quads From Hull. Creates a mesh of quadrilaterals that forms a polygonal object from the selected object. You will have to hide or delete the original object if you do not want it to show up in your scene.

Editing the NURBS Surface

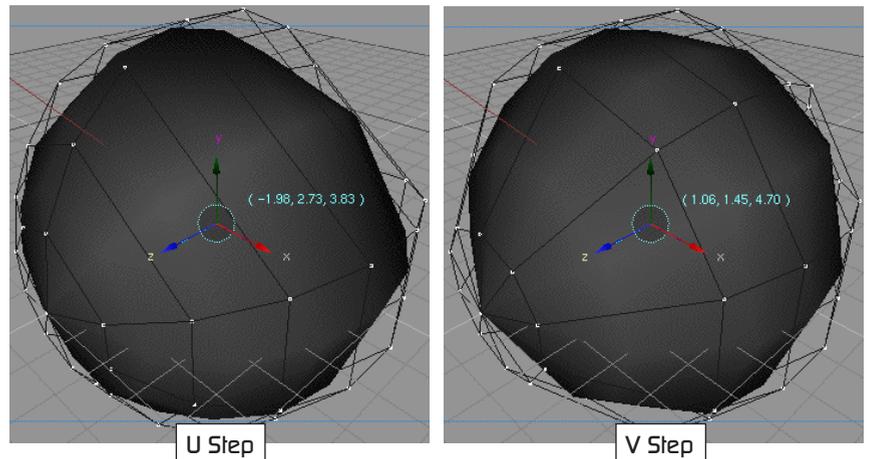
Insert>U Step/V Step. Inserts a column of vertices between the current vertex and the next vertex in the selected dimension (u or v). This menu option opens a dialog where you can specify a value between 0 and 1. This specifies where the new step will be inserted as a percentage of the way between the current step and the next step.

Figure 8.27: Inserted U Step and V Step



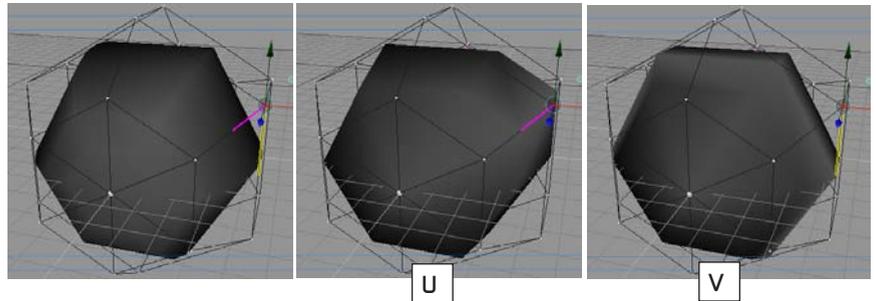
Remove>U Step/V Step. Deletes the currently selected u or v step.

Figure 8.28: Deleted U Step and V Step



Double>U/V. Doubles a u or v column, creating a crease or bump along those vertices.

Figure 8.29: Doubled U and V



Subdivide. Subdivides each u and v step into the specified number of steps. This opens a dialog where you can independently specify the number of u and v subdivisions.

Reorder>Patch/Spline. Reinterprets the current shape with the specified number of u and v steps. Reorder patch allows you to specify both u and v steps. Reorder spline only allows you to specify u steps.

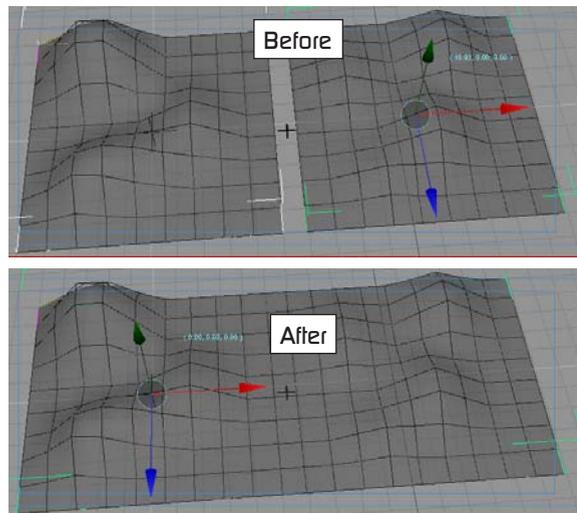
Simplify>U/V/Patch. Reduces the number of u, v, or u and v steps so that the rendered shape is approximately the same with the minimum number of steps.

Join>U to U/V to V. Joins the end furthest from the origin (in the specified dimension) of the first selected object to the end closest to the origin of the second selected object. Both objects must be selected before you choose join.

In the example above the mesh on the left was selected first and then the mesh on the right was selected (shift-click to add the mesh to the selection).

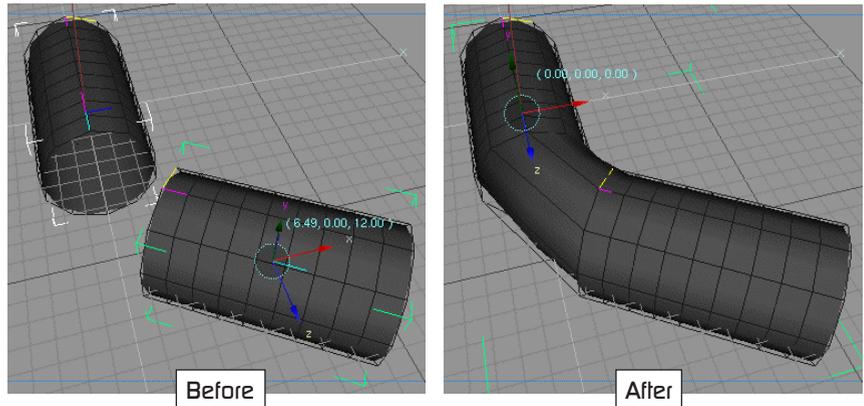
In the example below, the mesh on the left was selected, then the mesh on the right was selected (shift-click to add the mesh to the selection). If the meshes were selected in the opposite order the wrong ends would have been joined.

Figure 8.30: Joined U to U



In the example below the top left cylinder was selected first and then the bottom right cylinder was selected.

Figure 8.3 I : Joined V to V



Invert>U/V. Switches the origin and direction of the specified dimension so the origin gets moved to the side that it was furthest from.

Transpose. Swaps the u and v dimensions.

Shift Origin>U Back/U Forward/V Back/V Forward. Shifts the origin one step in the specified direction.

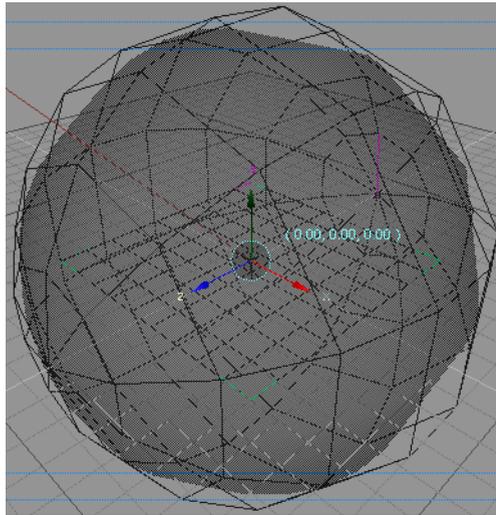
Close Ends>Sharp/Round *n*%. Brings the u vertices at each end together. In a cylinder-like shape this forms a kind of north and south pole.

Wrap

Wrap. Use this to wrap one surface to another.

To wrap a mesh to a sphere...

1. Create a NURBS mesh inside of a NURBS sphere.

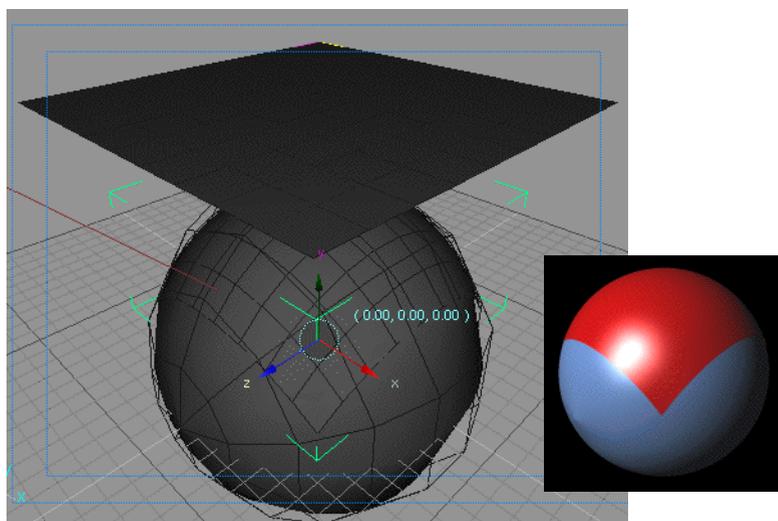


2. Select the mesh and choose **Duplicate** from the **Edit** menu to create a second mesh.
3. Move the second mesh above the sphere. You can also resize the second mesh if you want the projected mesh to be a different size.
4. From the menus select **Edit NURBS>Wrap**.
5. When prompted, select the source object. This is the mesh that will get projected onto the sphere. You can choose either mesh. In this example the inside mesh was chosen as the source.

6. When prompted, select the target object. This is the other mesh (the outside one).
7. When prompted, select the projection surface. This is the sphere.

This wraps the inside mesh across the surface of the sphere as if it were projected from the source position onto the sphere toward the target object.

Figure 8.32: Wrapped Mesh



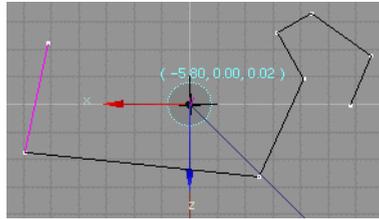
Wrap and Project. Use this to wrap a curve onto a surface. Unlike **Wrap**, **Wrap and Project** is dynamic. That is, if you alter the projection surface after you **Wrap and Project**, the curve will change shape with the surface. (See **Lofted Blends** on page 333 for an example using **Wrap and Project**.)

Curve Control

Degree

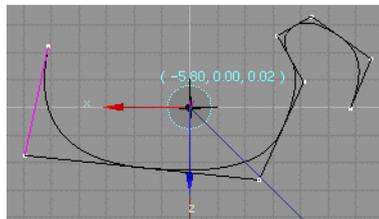
The *degree* of a NURBS surface or curve affects the overall shape. A **Linear** curve is a series of straight lines that go through each vertex.

Figure 8.33: Linear Curve



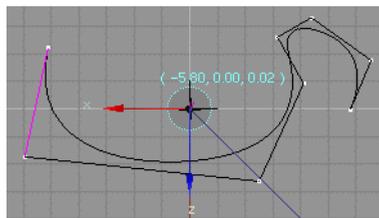
The default **Cubic** curve is tangential to the lines between the vertices.

Figure 8.34: Cubic Curve



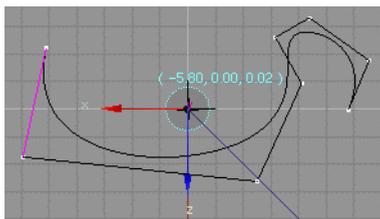
The **Quadratic** and **Quintic** curves are progressively looser.

Figure 8.35: Quadratic Curve



Select

Figure 8.36: Quintic Curve



To change the degree of a NURBS surface or curve...

1. Select the NURBS object or curve.
2. Select degree you want from the **Edit NURBS>Degree>U** menu or **Edit NURBS>Degree>V** menu. (Curves created from the **Curves** menu only have a u dimension, so you can only change the u degree of a curve, while you can change either dimension of a NURBS object.)

Knots

Knots are values that control how much influence vertices have on the shape of the curve or surface. There are two settings you can choose from the **Edit NURBS>Knots>U** and **Edit NURBS>Knots>V** menus.

Semi-Uniform. The knot values are set so the end points have a looser, interpolated fit.

Uniform. The knot values are set at equal increments creating a tighter fit at the end points.

Select

Use the **Select** menu options to move the selection one step forward or backward in the selected dimension.

Attributes

If you open an object in the Attribute Manager by clicking on the triangle to the left of its name, you will see several attribute categories. You can edit any object numerically (rather than editing it in the views) by changing its attributes.

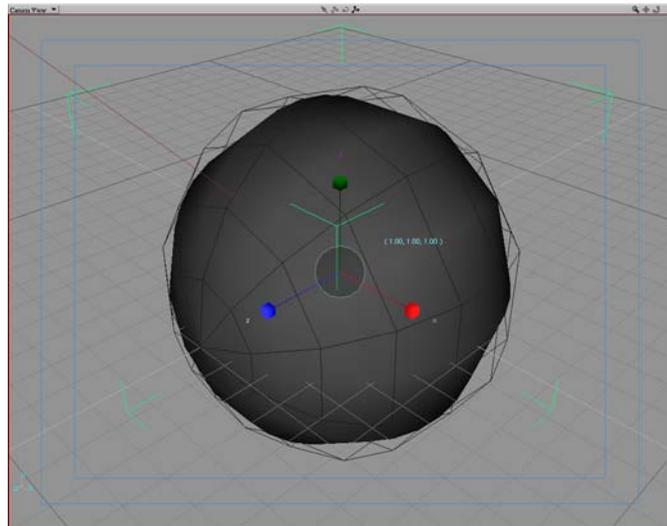
Transformation	These attributes specify whether the object has been moved, resize, rotated, or whether the object center has moved relative to the object. To reset the values so the current location, size, and rotation becomes the object's start location, size, and rotation, choose Park from the Control menu.
Render Options	These attributes specify the shader attached to the object and control how the shader is rendered. A shader is what gives an object a certain appearance. For more details about shaders, see Shading on page 165.
NURBS	Shows the number of <i>u</i> and <i>v</i> steps and whether the object is periodic in the <i>u</i> or <i>v</i> dimension. (For more information on periodicity, see Periodic NURBS, U, and V on page 39).
Display	The display settings affect how objects appear in the views. Display settings do not change how objects are rendered. In general, the more detailed and accurate the display, the slower it is to refresh.
Morph Targets	To animate things like facial expressions you distort your object into a pose and save that pose as a morph target. When animating you can then mix poses in different percentages. For instance, you can save a happy face as a morph target, and you can save the same face making an "oo" sound as another morph target. Then in your animation you can mix the two so that your face looks like it is happily making an "oo" sound. Morph targets will be described in more detail in the chapter about CharacterAnimation Tools on page 319.

Attributes

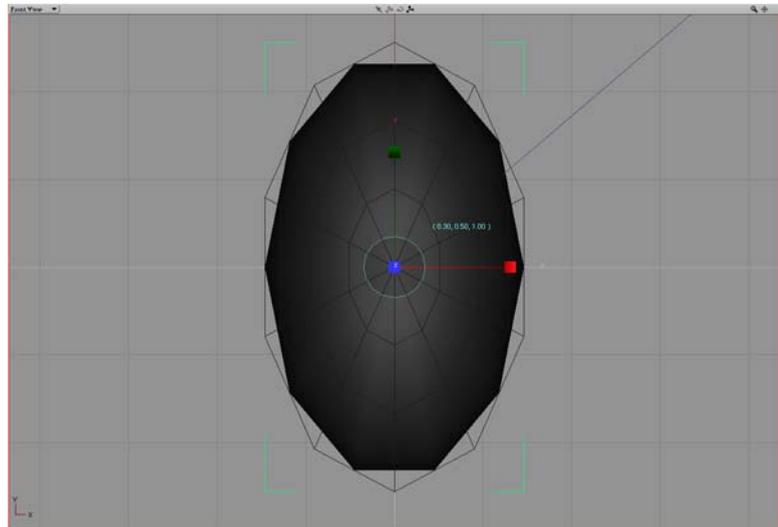
Chapter 9. Tutorial: Modeling a Molly

In general, the way you create something in a modeler like PiXELS 3D is you start with a simple shape and reshape it into something. In this case, you will start with a sphere and reshape it into a fish's body.

1. Click on the **Sphere** icon in the toolbar to create a sphere. When you click the icon, a dialogue box opens. For now, accept all the default values and click **OK**. The sphere appears in each of the four views.
2. Next, you need to squash the sphere so that it is thinner, more like a fish's body. Click on the scale icon above one of the views. Of the four icons centred above each view it is the left-most icon. The red, green, and blue arrows pointing away from the centre of the sphere change into colored lines with colored boxes at the ends. These lines with boxes are called *scale handles*.



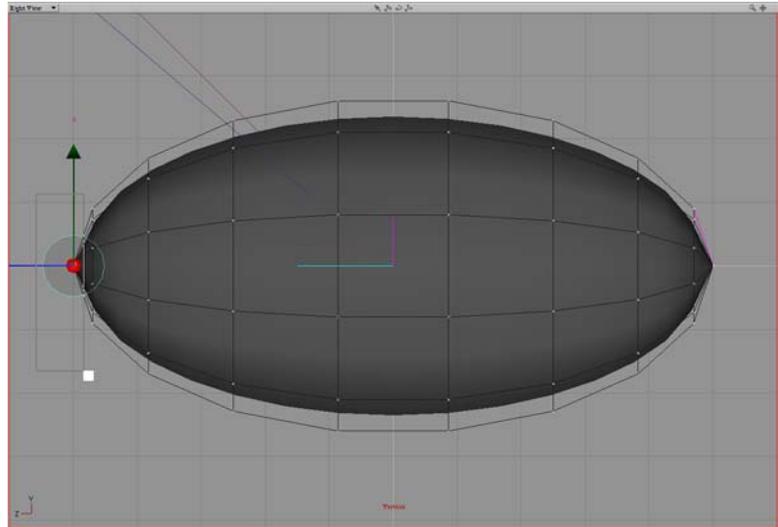
3. In the **Front View** click and drag the red scale handle marked **x** toward the centre of the sphere. When you click on the handle, the line turns yellow to show it is active. As you drag the handle this flattens the sphere into a disk.
Notice how the **Front View**, **Top View**, and **Camera View** show how the sphere changed.
4. In the same view drag the green scale handle marked **y** toward the centre of the sphere. This squashes the disk into a sh-like shape.



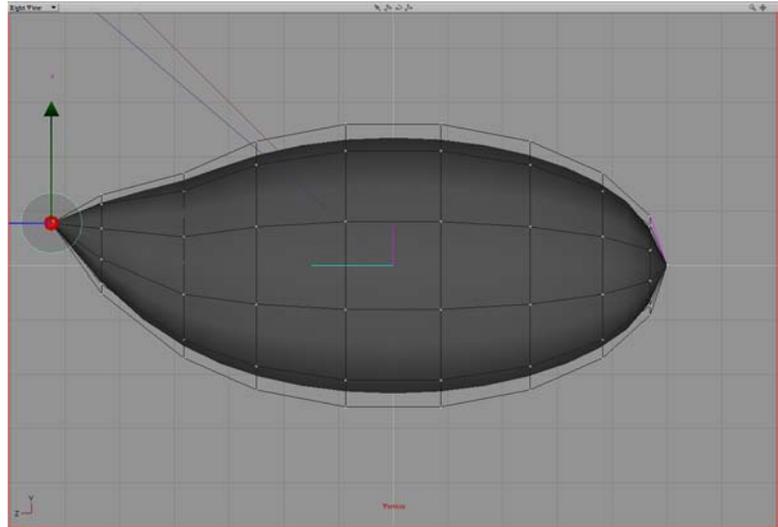
Next, you will change one end of the shape to start the head.

1. From the menu bar select **Edit NURBS > FreeForm**. This selects the **FreeForm** tool and switches you from **Object** to **Vertex** editing mode.
2. In the upper right corner of the screen is the **Tool Manager**. You should see the word **FreeForm** at the top of this window. The first item is **Mode**, with a drop-down list. Make sure the **Mode** is set to **Pull**.

3. Select the front end of the shape's body by clicking near the end point, then dragging across the end point. (Your cursor will draw a rectangle on the screen.) This is important, since what you see as one end point is actually many points at the same place. The click-and-drag select will select all of the points.

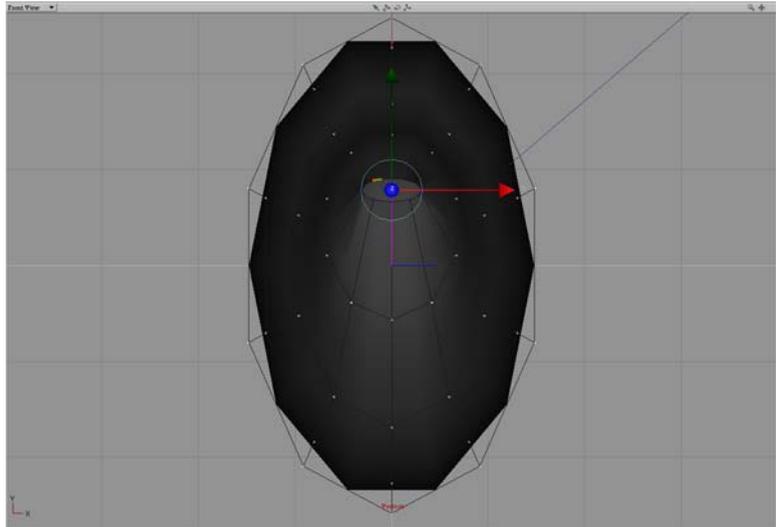


4. In the **Right View**, click within the grey halo around the end points, and drag the cursor to move the end of the shape up and to the left.



Next, you will give the fish a mouth. You will want to work in the **Front View** for this.

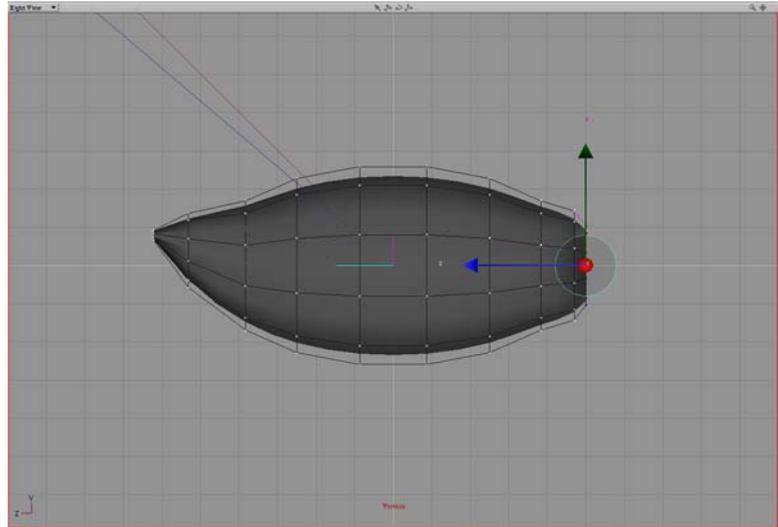
1. In the **Tool Manager**, change the **Mode** from **Pull** to **Pinch** by selecting **Pinch** from the first drop-down list.
2. In the **Front View**, click on the red **FreeForm** handle marked **x** and drag the cursor to the right to widen the mouth.
3. In the **Front View**, click on the green **FreeForm** handle marked **y** and drag the cursor downward to open the mouth a bit.



In the above steps, make sure you click on the part of the handle outside the grey halo. If you click inside the grey circle, you will be working in more than one direction, and you will get a strange-looking shape.

Next you will reshape the other end of the shape's body so you can attach a tail to the shape.

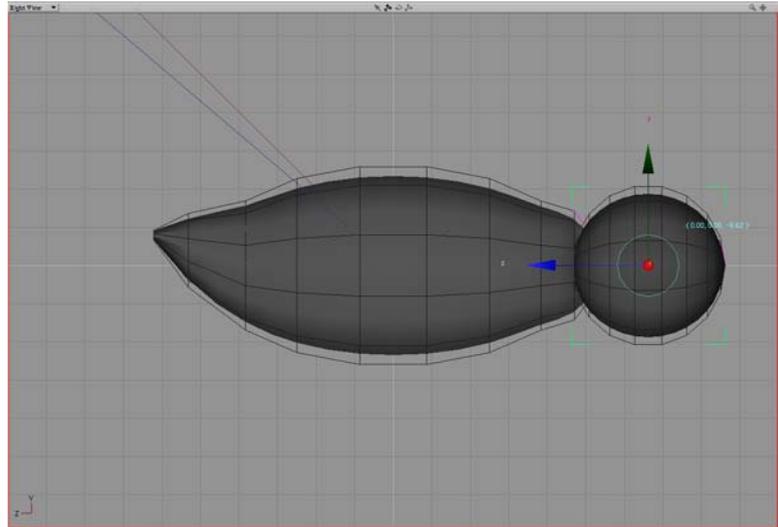
1. In the **Right View** click-and-drag to select the back end of the body shape.
2. Click on the green **FreeForm** handle marked **y** and drag downward to widen the section where you will attach the tail.



When you are done reshaping the body, exit **FreeForm** editing mode by selecting one of the four tools above a view and selecting **Object** from the view contextual menu. (You open the view contextual menu by Control-clicking in a view.)

Now you can create the tail. You will create another sphere for the tail, but this time you will take a short-cut to start with a smaller sphere.

1. Open the **Sphere** dialogue by clicking on the **Sphere** icon in the toolbar. In the **Radius** field on the dialogue enter **2** and press the return key, then click the **OK** button.
2. In the **Right View**, click the **Move** icon above the view. (The **Move** icon looks like three lines with arrows at the ends. It is the second of the four icons centred above each view.) Click-and-drag the blue **Move** handle marked **z** toward the back end of the fish so it just overlaps the end of the body.

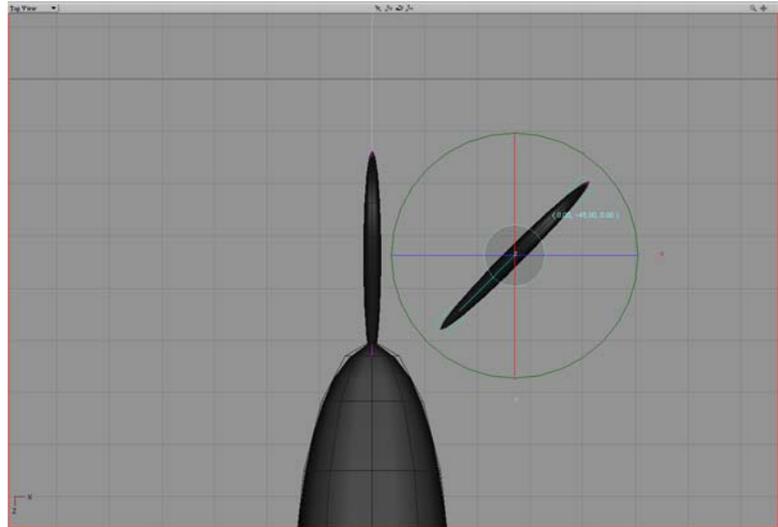


3. In the **Top View**, click the **Resize** icon (three lines with squares at the ends). Then click on the red **Scale** handle marked **x** and drag the cursor to the left to squash the sphere until it is almost at.
4. From the menu bar select **Edit NURBS > FreeForm**. You should still be in **Pinch** mode. Select **Mirror U** and make sure the **Mirror Axis** value is set to **X**.
5. In the **Right View**, click on one of the control vertices (a white square on the black line or wireframe) around the tail, about a third of the way from the right edge of the tail and on the bottom edge. Drag the green handle marked **y** downward to broaden the tail at that point.

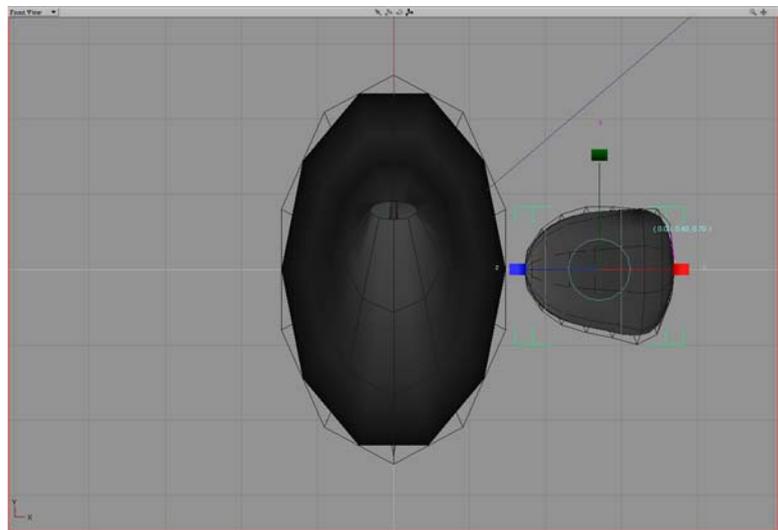
You may want to select and **FreeForm Pinch** another point on the wireframe to make the curve a bit smoother. Remember to use the handles rather than the grey halo when you **FreeForm Pinch**, otherwise you will make the tail fatter. When you are done, go back to **Object** mode by selecting **Object** from the view contextual menu.

Next, you will add a pectoral fin on the side of the fish.

1. With the tail selected choose **Edit > Duplicate** from the menu bar. Nothing will appear to happen in the view. Look in the **Attribute Manager** (the list which starts with **Objects** underneath the **Tool Attributes**). Open the **Objects** attributes by clicking on the open triangle next to the word. At the end of the listed **Objects** there are three **Spheres**. The first is the body. The second is the tail. The third is the copy of the tail that you just made.
2. Click on the **Move** icon above a view and use either the red **Move** handle marked x or the grey halo to move the new fin away from the tail.
3. In the **Top View** click on the **Rotation** icon (the third of the four tool icons). Rather than lines with a symbol attached to the ends, the **Rotation** handles look like colored circles.
4. Click-and-drag the green **Rotation** handle to the right until the fin is approximately at a 45 degree angle to its original position. The three numbers above and to the right of the grey halo should read something close to **0.00, -45.00, 0.00**.



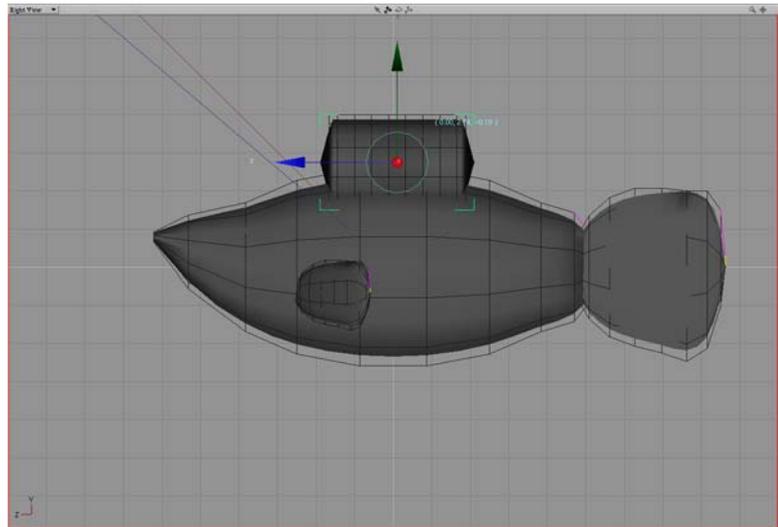
5. In the **Front View**, click the **Scale** icon. Drag the green scale handle marked **y** down, then drag the blue scale handle marked **z** up. Both operations make the **n** smaller in one dimension.



6. In the **Right View**, click the **Move** icon, then click-and-drag the grey halo around the centre of the **n** to move it so it looks like it is in the right place on the side of the **sh**. In the **Top View**, move the **n** so that it is partly inside the **sh**'s body

The **sh** also needs a top, or dorsal, **n**. Since the dorsal **n** is long and not-quite-rectangular, you will use a cylinder to start with rather than a sphere.

1. From the **NURBS** menu, select **Cylinder**. When the **Cylinder** dialogue opens, change the **Radius** to **1.25** and the **Length** to **4**. Initially the cylinder will be inside of the **sh** body, so you will only be able to see the handles, not the cylinder itself.
2. Move the cylinder to the top of the **sh**'s body
3. From the **Edit NURBS** menu, select **Close Ends > Round 50%** to make the cylinder solid-looking.



4. In the **Top View**, squash the fin along the x axis by clicking on the **Resize** tool and then click-and-dragging the red Size handle marked x.

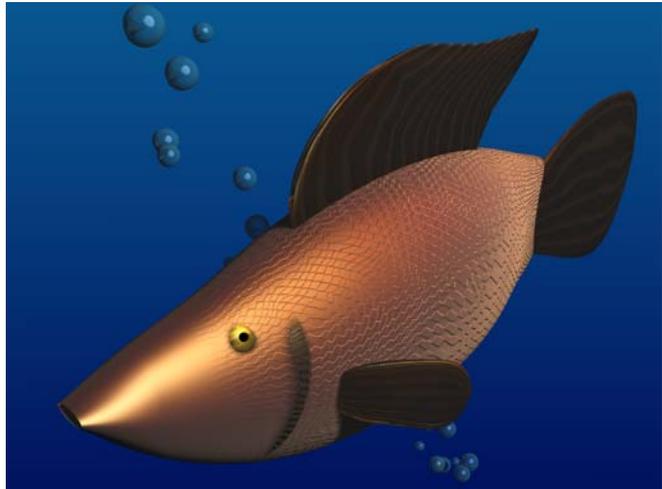
Now that you have a basic fin shape to start with, you will want to refine the shape to make it look more like a fin.

1. Select **Edit NURBS > FreeForm**. In the **Tool Manager**, make sure the **Mode** is set to **Pull**.
2. Click and drag to select the top row of points on the fin.
3. Click inside of the dotted green circle and drag the top edge of the fin towards the tail.
4. Click on the background to de-select the points.
5. Click and drag to select the top corner of points and drag them further towards the fin.

At this point, you may see some sharp edges and points along the other edges of the fin. You can select the points and move them to smooth the overall shape of the fin.

To add further details to the fish, you could add an eye, additional fins on the underside of the body, a gill flap, or some texturing and shading to make the final render look better. Most of the pieces of the fish could be modelled with different

methods, starting with different shapes or using different tools.



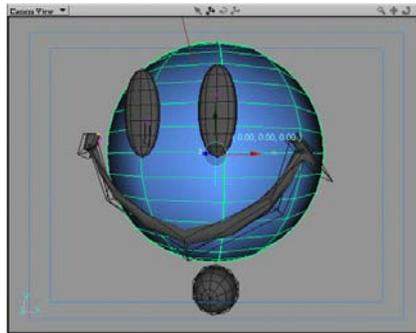
Chapter 10. Tutorial: Modeling a Face

In the following tutorial you will use NURBS shapes to create a simple cartoon head. For very simple characters you can often create the face from scratch without any kind of precise template. For more complex or realistic models you will want to use a background image as a template.

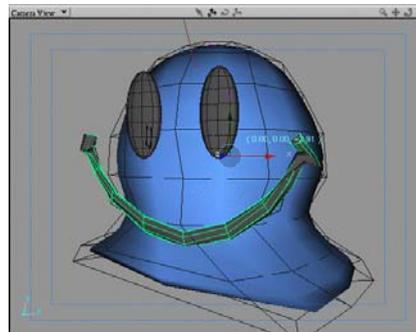
Create the Nose

There are many ways of creating a face or head. Most NURBS techniques are based on either a sphere or a hemisphere.

Sphere-Based Head

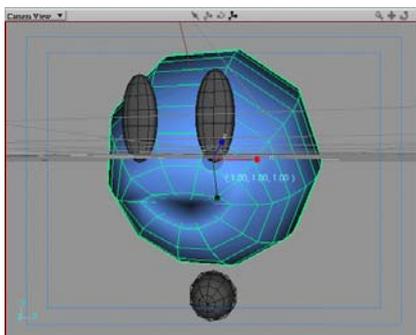


Hemisphere-Based Head



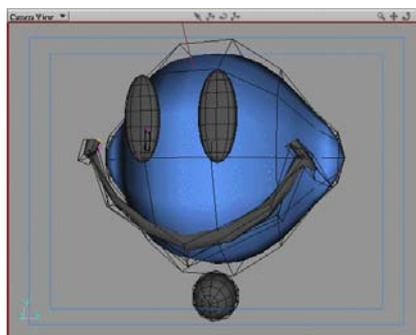
The main difference between different techniques is the location of the sphere's axis. If the character's mouth is going to open and close, you will usually find it easier to place the axis at the center of the mouth.

Axis At Mouth

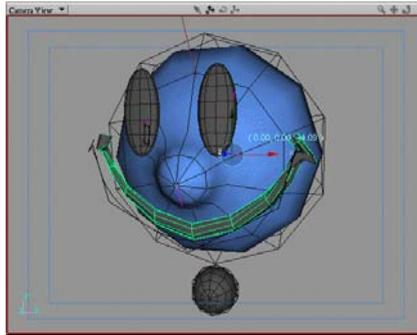


Otherwise, you can place the axis anywhere you want. The ears, the top of the head, or the nose are all good places for the axis.

Axis at Ears



Axis at Nose



In this tutorial you will place the axis at the center of the nose. You will start by forming the nose out of a cylinder.

Create A NURBS Cylinder

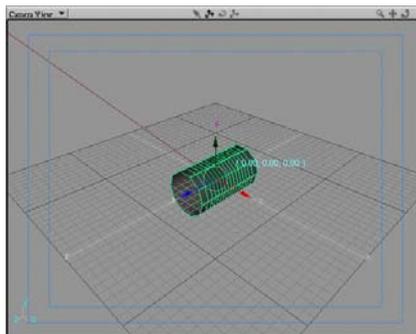
1. Select **Cylinder** from the **NURBS** menu. This opens the **Cylinder** dialog.

Cylinder Dialog



2. Click **OK** to accept the defaults. This creates a cylinder as shown below.

NURBS Cylinder

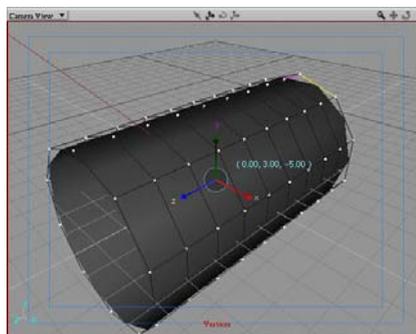


3. Click and drag on the  (Zoom) icon or hold the Option key and drag the mouse to zoom in on the cylinder in the Camera View.

Pinch the End of the Cylinder

1. Select the FreeForm tool from the Edit NURBS menu. This changes the views to Vertices mode.

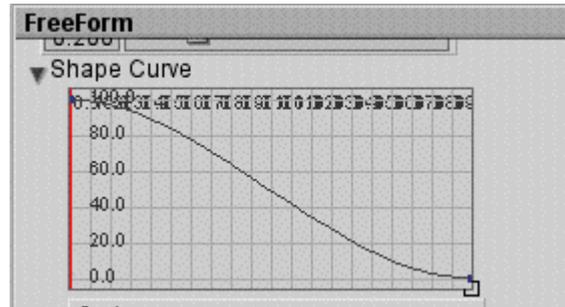
FreeForm Tool Selected



2. In the Tool Manager (the list at the right titled FreeForm) change the Mode from Pull to Pinch.

Since you want the end of the nose to be rounded, you will use the Shape Curve attribute to control how the Pinch looks.

3. Open the **Shape Curve** attribute in the **Tool Managers** list by clicking on the triangle () by the attribute name. (You probably will have to scroll down to see the whole Shape Curve.)

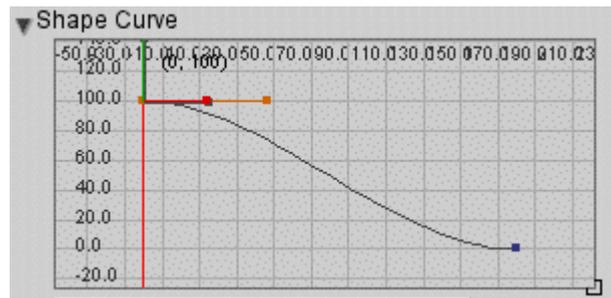


4. You can click and drag in the **Shape Curve** graph to move the view of the curve. To zoom in or out, hold down the **Option** key and click and drag in the graph. To resize the graph, grab the handle in the lower right.

Adjust the view of the graph so you can see both ends of the curve.

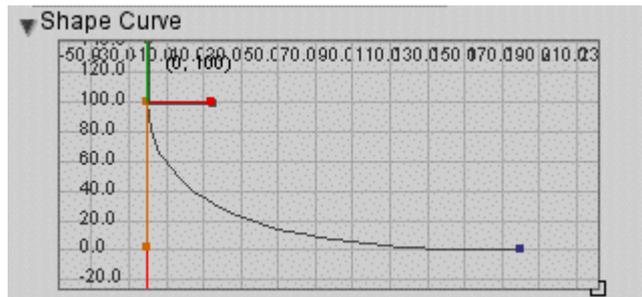
5. Click on the upper left end point on the **Shape Curve**. With the end point selected you can see orange control point handles.

Orange Control Handles



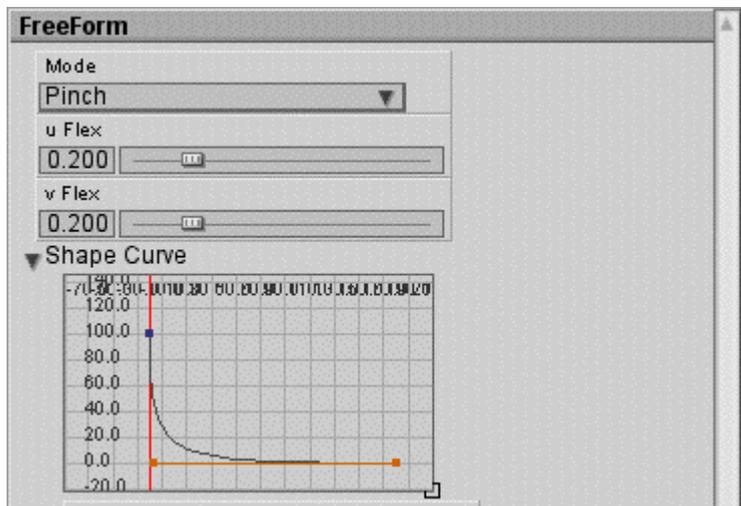
6. Drag the orange control point down to **0.0** directly underneath the end point.

One Handle Adjusted



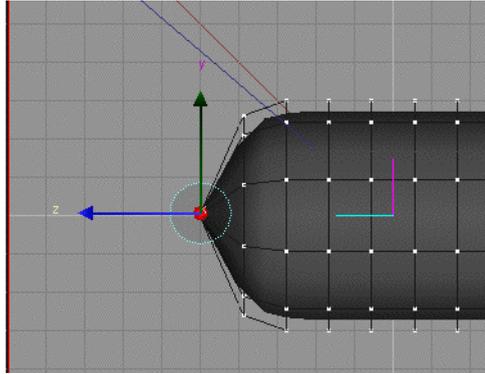
7. Select the other end of the **Shape Curve** and drag the control point horizontally to approximately the same place. The resulting **Shape Curve** is shown below.

FreeForm Attributes



8. In the **Right View** select a vertex on the left end of the cylinder.
9. Click and drag the mouse up to pinch in around the cylinder at this point. Drag until all the u vertices meet. The resulting shape is shown below.

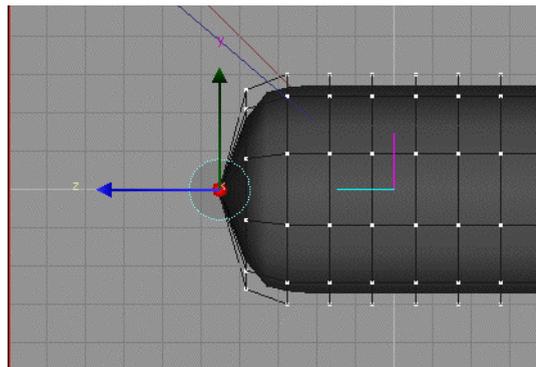
Pinched Cylinder



Pull the End of the Cylinder

1. Deselect the vertex by clicking in the background of the view.
2. Select all the vertices at the point of the nose by clicking and dragging a square around the point.
3. Now change the **FreeForm Mode** back to **Pull** and drag the vertex along the **Z** (blue) axis to about halfway between the original position and the next u column of vertices.

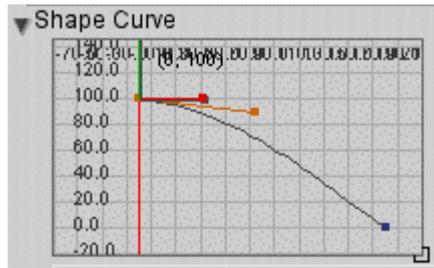
Pulled Vertex



Expand the Middle of the Cylinder

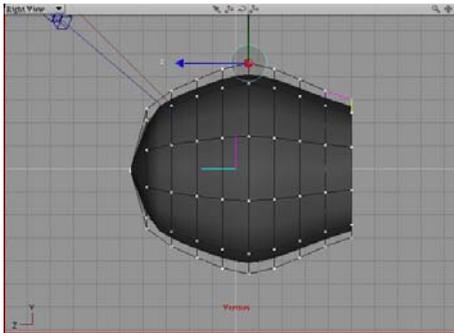
1. Now use the orange handles on the **Shape Curve** to make the curve nearly diagonal with a slight bulge near the upper left as shown below.

New Shape Curve



2. Change the **v Flex** attribute to 0.4 and change the **Mode** back to **Pinch**.
3. Now select a point near the middle of the cylinder and drag down to make the cylinder look like a bulbous cartoon nose.

A Cartoon Nose



You can give the nose a slightly rounder shape by selecting a vertex in the third or fourth column from the left and dragging down a little bit.

Save Your Work

Save what you have done so far in a file called **CartoonHead.piv**.

When you start to do more complex files, you will want to save in a different file before each major change. That way you can always go back to a previous version if you get your scene hopelessly messed up.

If you are using PiXELS in a professional environment, consider using a version or file history control system like the freely-available MacCVS (www.maccvs.org). On OS X you can also use the command line CVS that comes with the Developer Tools. Aladdin also has a commercial file history system called Fastback. If you regularly merge your changes back into the file history database, you can very easily go back to previous versions of your scene.

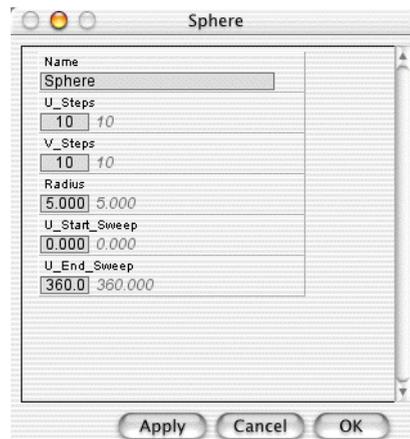
Create the Head

Create a NURBS Sphere

Once you have finished the nose, you are going to make the rest of the head from a sphere.

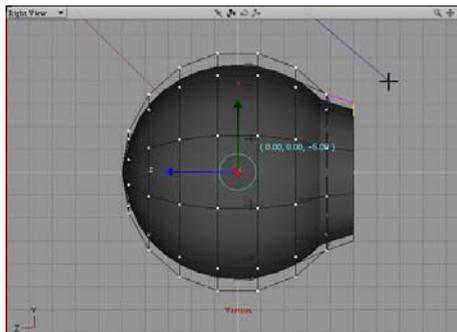
1. In the favorites bar, click on the NURBS Sphere icon () . This opens the Sphere dialog.

Sphere Dialog



2. Click **OK** to accept the defaults. This adds a new sphere to your scene.

Sphere Added



3. To switch out of **FreeForm** mode, click on the **Move** () icon at the top of a view.
4. Now **Control-Click** in a view to open the **View** contextual menu.

View contextual menu

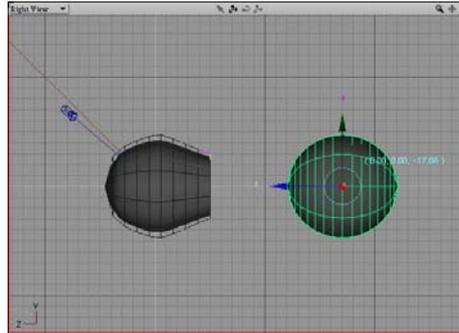


5. Click on **Object** to switch from **Vertices** to **Object** mode.

Move and Scale the Sphere

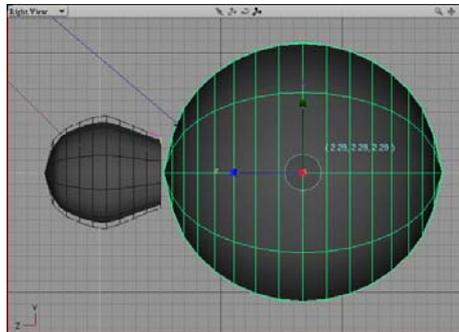
1. Move the sphere back along the **Z** (blue) axis until its center is a little more than one nose length behind the nose.

Reposition Sphere



2. Click on the **Scale** () icon.
3. Click in the grey circular object handle (where the three colored scale handles meet) and drag the mouse to the right to expand the sphere. Resize the sphere until its surface almost touches the nose.

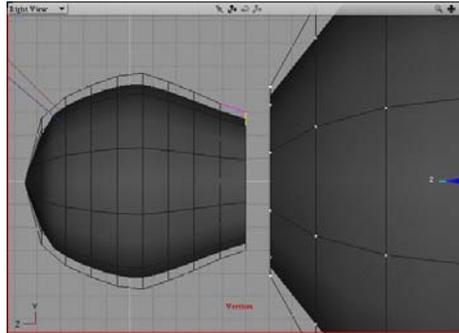
Resized Sphere



Connect the Sphere

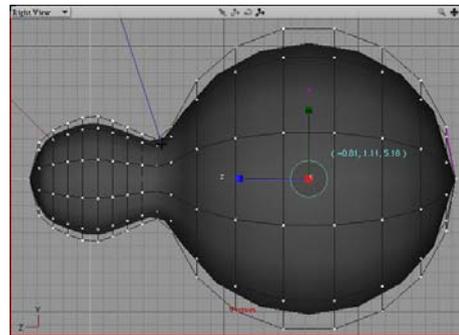
1. Switch back to **Vertices** mode by pressing the **F10** key.
2. Select the vertices at the tip of the sphere closest to the nose.
3. From the menus select **Edit NURBS>Remove>V Step**. This cuts a circular hole in the sphere.

V Step Removed



4. Press **F9** to switch back to **Object** mode and either drag the mouse or hold down the Shift key and click on each object to select the sphere and the nose. *Make sure you select the sphere first and then select the nose.*
5. From the **Edit NURBS** menu select **Join>V to V**. This connects the cutout circle to the back of the nose.

Nose Joined to Sphere



Save Your Work

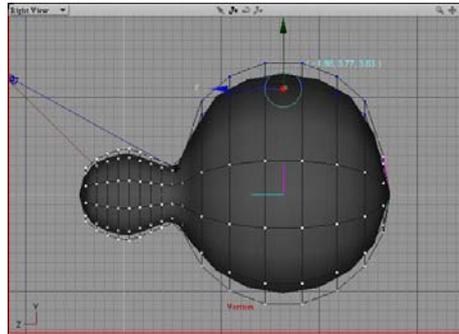
Save what you have done so far in **CartoonHead.piv**. This basic shape is a good starting point for a number of different cartoon heads.

To keep this shape, start another file by saving it as **CartoonHead2.piv**. In the next steps where you reshape the head, you will save your changes in this new file.

Reshape the Head

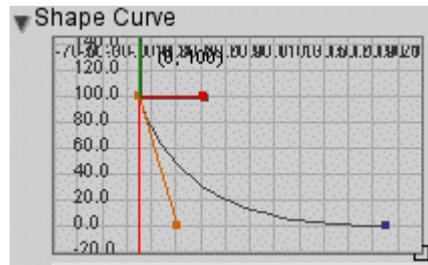
1. Select the vertices at the top third of the head and move them up a little.

Head Top Stretched



2. From the **Edit NURBS** menu select **FreeForm**.
3. In the **Tool Manager** list change the shape curve to a shallow cup.

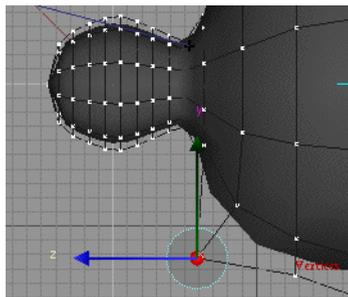
Shape Curve



4. Change the **u Flex** and **v Flex** to **0.07** and change the **Mode** to **Pull**.
5. Pull the vertex at the bottom front of the sphere (where the chin is) out a little.

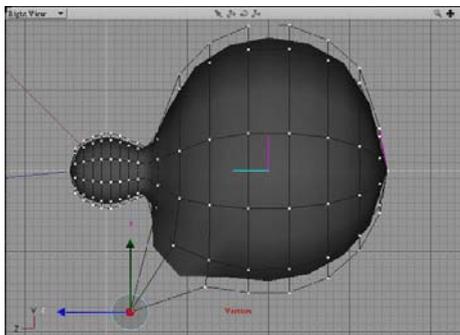
Reshape the Head

Chin Pulled Out a Little.



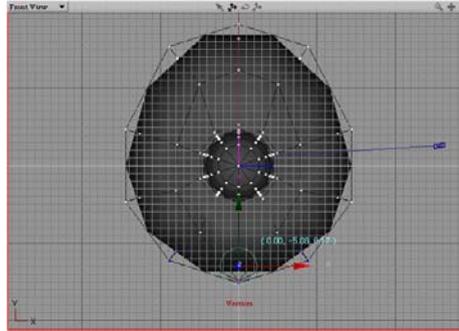
6. Change the **u Flex** and **v Flex** to **0.2** and pull the chin some more.

Chin Pulled More



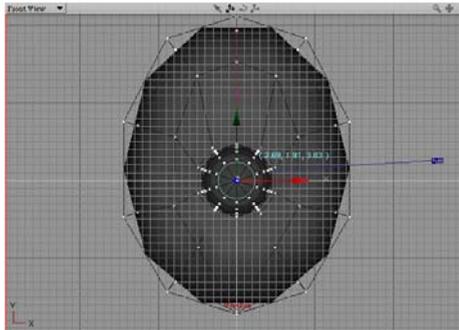
7. In the **Front View** select the vertices at the bottom of the head.

Bottom Vertices Selected



8. Pull the vertices down to elongate the head a little.
9. Select the vertices at the top of the head and pull them up to elongate the head some more.

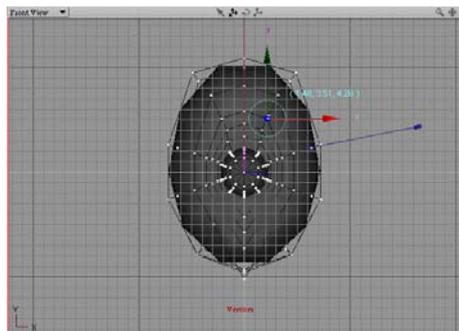
Elongated Head



Add Eye Sockets

1. Select one of the three points on the forehead and choose **Edit NURBS>Insert>V Step** from the menus. In the dialog that opens, accept the default and click OK.
2. Select the new point below the currently selected point and choose **Edit NURBS>Insert>V Step** again.

V Steps Added

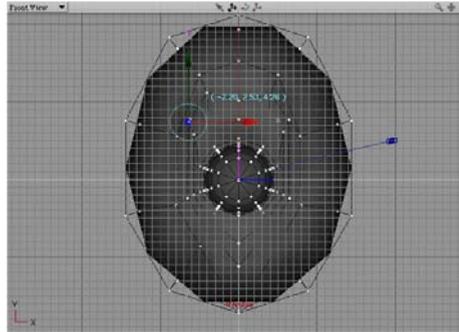


3. In the **Front View**, select the newly added vertices near where the eyes will be. (Use shift-click to select multiple vertices.)
4. In the **Right View**, move the selected vertices towards the back of the head to form eye sockets. The surface of the head will cover the vertices so you will no longer be able to see those points.

Note: Depending on how your cartoon head is shaped, slightly different techniques may work better. Here is an example of another approach that worked:

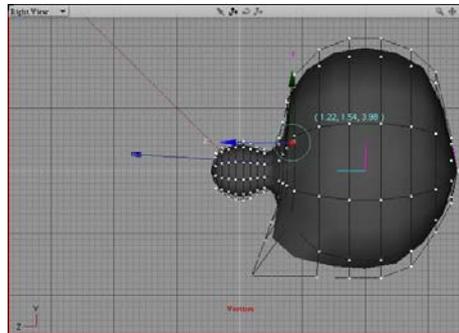
In the **Front View** reposition the newly added vertices so the ones in the top half of the face form contours around where the eyes will be. When you drag the points below the eyes down, the surface of the head will cover the vertices so you will no longer be able to see those two points.

New Points Repositioned



5. Drag the cursor across the area around (or below, if you used the second technique listed above) an eye socket to select one of the hidden points.
6. In the Right View drag the point toward the back of the head a little.

Eye Socket Formed



7. Select and drag the other eye socket point back to approximately the same Z position as the first. Do not worry about being exact. Slight irregularities will make the face look more natural.

Save Your Work

Once again, save your work, then create a new file to work from called `CartoonHead3.piv`.

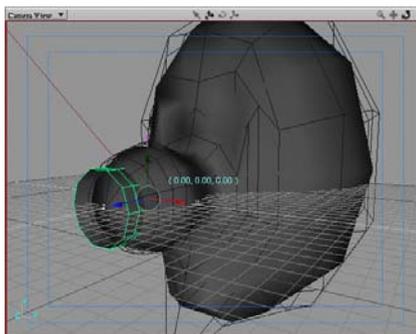
Add Eyebrows

Most of the facial expression for this character is going to come from the nose, the eyebrows, and the beard. The eyebrows are going to be extra big so you can see them under the big floppy wizard hat we are going to put on the character.

Create a Cylinder

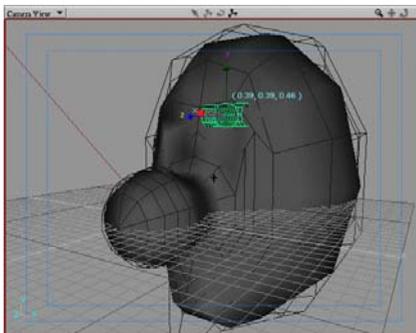
1. Press the **2** key to create a NURBS cylinder. This opens the Cylinder dialog. Click **OK** to accept the default.

Eyebrow Cylinder



2. Move, rotate, and resize the cylinder so that it forms a crude eyebrow.

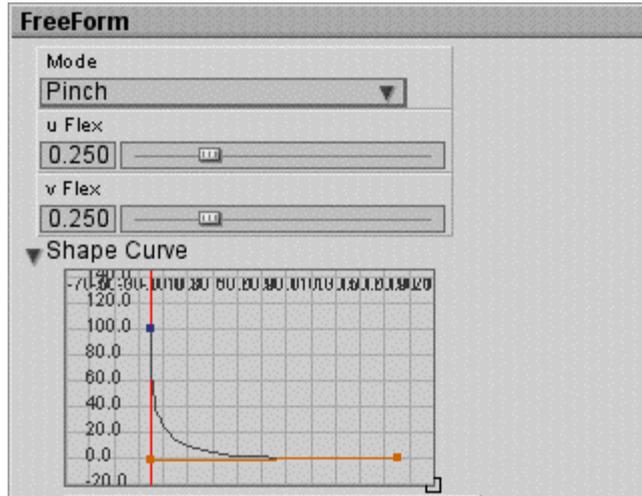
Crude Eyebrow



Shape the Eyebrow

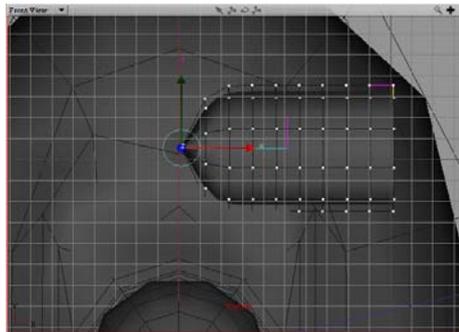
1. From the menus select **Edit NURBS>FreeForm**. Then adjust the **FreeForm Tool** attributes as shown (**Mode** at **Pinch**, **u Flex** and **v Flex** at **0.25** and a sharply cupped **Shape Curve**).

Adjust the FreeForm Tool Attributes



2. Select an end point on the cylinder, click in a view and using the grey circular handle drag the mouse up to pinch the end shut.

Eyebrow End Pinched Shut

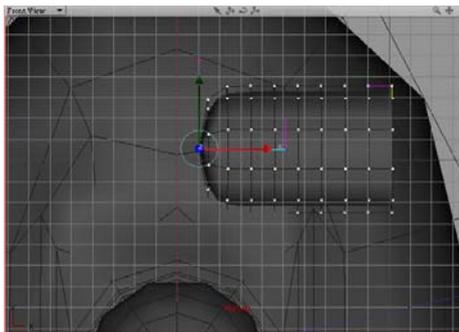


3. Now switch the **Mode** from **Pinch** to **Pull** and drag the mouse from outside the grey circular handle to just cover the tip of the cylinder so that all the points that have been brought together are selected.

Add Eyebrows

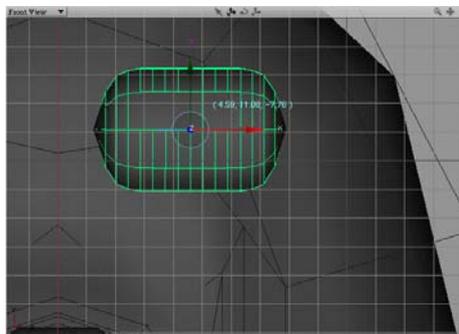
4. Drag from within the grey circular handle to pull the end points a little toward the center of the cylinder, blunting the end.

Blunted End of Eyebrow



5. Now repeat the last two steps (Step 3 and Step 4) with the other side of the eyebrow. (Note: When you are pinching the other side, grab a vertex at the bottom of the cylinder. The vertex at the top with the magenta line to the left of it is the origin of the shape. This vertex will not influence the other vertices when moved.)

Eyebrow Finished at Both Ends

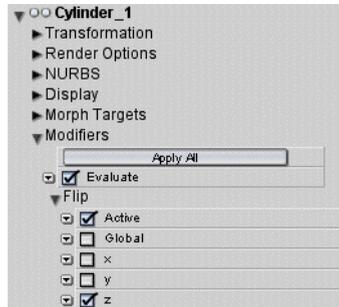


Create the Other Eye

1. Select **Duplicate** from the **Edit** menu to create the other eyebrow, then select **Edit NURBS>Modifiers>Flip**. This adds a **Flip** modifier to the currently selected eyebrow (which is currently named **Cylinder_1**).

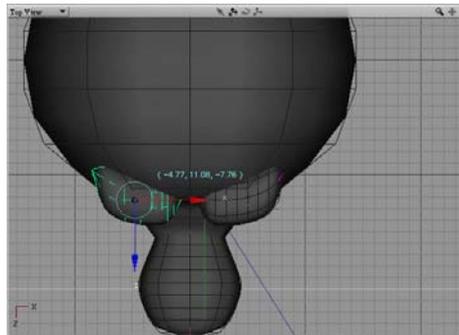
2. In the Attribute Manager click the arrow beside **Cylinder_1** to open the object's attributes. In the **Modifiers** attribute list there will be an item for **Flip**. Enable the **z** attribute under **Flip**. This flips the eyebrow so it is now a right eyebrow rather than a left eyebrow.

Enable Flip z



3. Move the new eyebrow over the character's right eye socket (the one on your left).

Two Eyebrows



Save Your Work

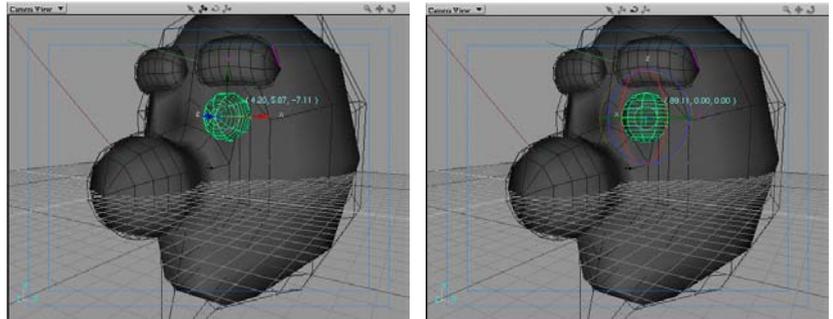
Once again, save your work, then create a new file to work from called **CartoonHead4.piv**.

Add Eyes

1. Add a sphere for the left eye. After moving it into place and resizing it, rotate it 90° around the x-axis. In this orientation it will be easier to use an image map for the pupil.

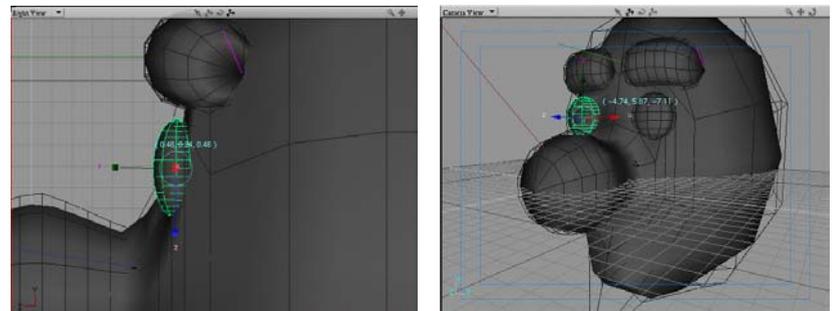
Add Beard

Sphere for an Eye



2. Scale the eye along the y axis to make it flatter, then duplicate it. Move the second eye over so it becomes the right eye.

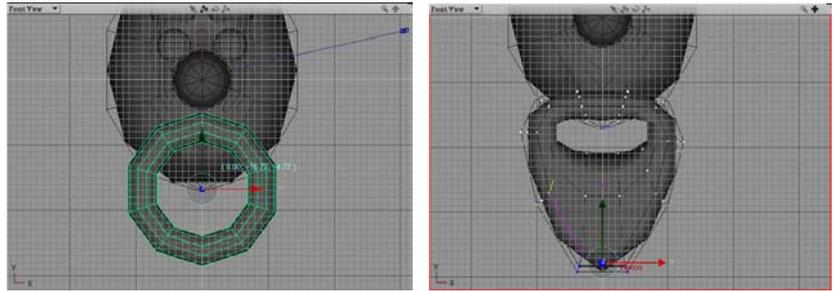
Flatten and Duplicate Eye



Add Beard

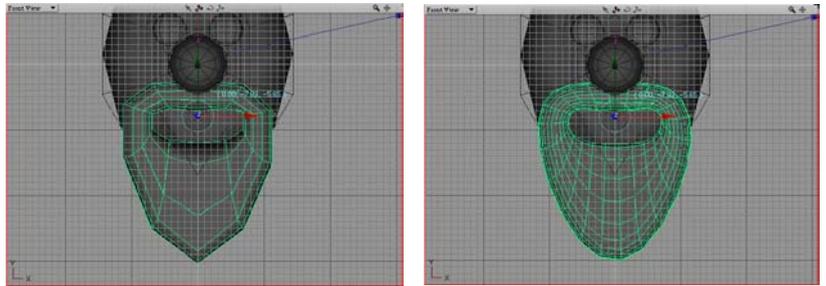
1. Create a torus for the beard, then *FreeForm Pull* the top of the torus down to flatten it, pull the inside bottom vertices up a little, and pull the outside bottom vertices down.

Create and FreeForm Pull Torus



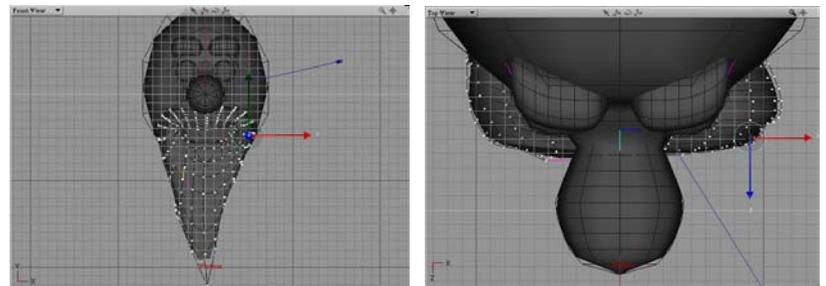
2. Move the beard back into position, then select **Edit NURBS>Subdivide**.

Reposition and Subdivide Beard



3. **FreeForm Pull** the beard to reshape it in the front and top views.

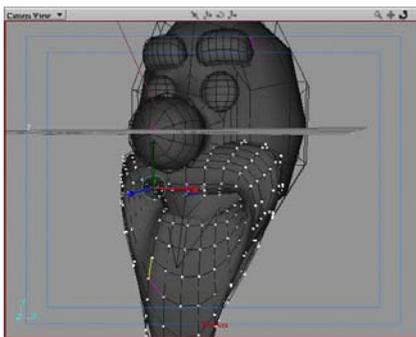
Reshape Beard



Add Hat and Hair

4. Do some final adjustments to close up the gaps between the beard and the face using **FreeForm Pull**. These gaps will occur mostly in the mouth area.

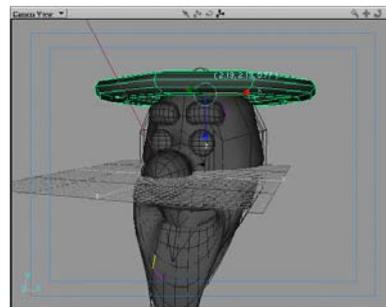
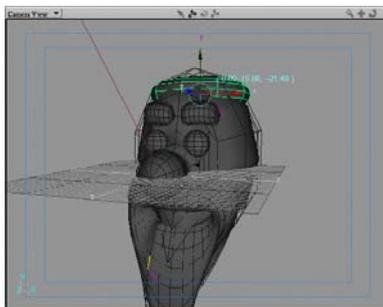
Final Adjustments



Add Hat and Hair

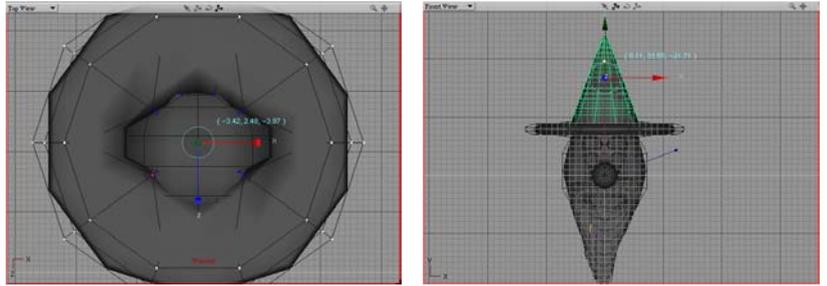
1. Add a torus for a hat brim, then scale it along the x and y axis.

Add a Torus and Scale It



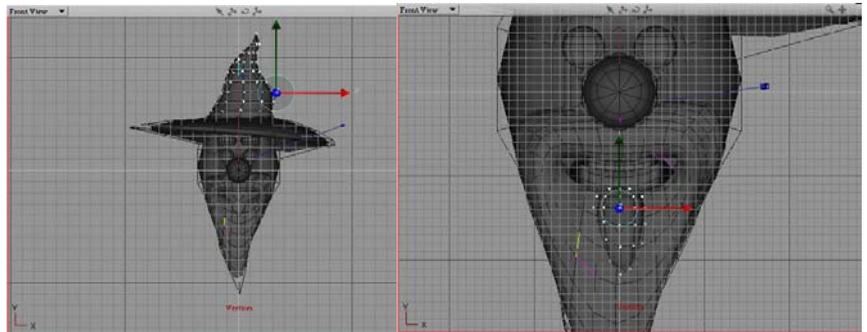
2. Drag the points on the inside of the torus toward the center in the **Top View**, then add a cone for the top of the hat.

Pull Inside of Brim Toward Center Then Add Cone



3. Use **FreeForm Pull** to adjust the hat, then add two cylinders for lips and a sphere for beard hair under the lips. Use **FreeForm Pull** to shape them.

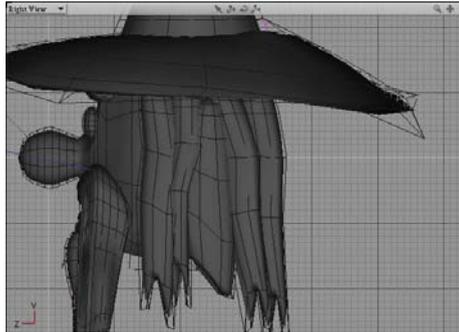
Make Hat Irregular, Add Lips, and Hair Under Lips



4. Create spheres or cylinders with their ends closed for the hair. Use **FreeForm Pull** to make each hair object a little different.

The Finished Wizard

Add Hair



The Finished Wizard

After you have saved your work, do a test render and make final adjustments to the shape.



Once the shape is complete, add shading and lights.



The Finished Wizard

Chapter 11. LogoMaker

Getting Started

LogoMaker lets you create 3D text, 3D logos, or other complex objects based on extruded 2D artwork.

Creating the Text

With LogoMaker you can either start with splines drawn in PiXELS or you can start with artwork drawn in an illustration package like Adobe Illustrator or Macromedia Freehand.

In Freehand 10.

1. Type any text you want in the logo.
2. Select the text.
3. Select **Text>Convert to Paths**.
4. Draw the outline of any objects you want in the logo. The outline must be a closed path. You can cut out areas of the interior by drawing closed paths inside the main outline.
5. Select all the curves that form an outline and make the stroke color none and the fill color black.
6. Select **Modify>Join** to make the object into a single curve.
7. Select **File>Export**.
8. In the **Export** dialog, select **Adobe Illustrator 5.5**.

In Illustrator 10.

1. Type any text you want in the logo.
2. Select the text.
3. Select **Type>Create Outlines**.
4. Draw the outline of any objects you want in the logo. The outline must be a closed path. You can cut out areas of

the interior by drawing closed paths inside the main outline.

5. Select all the curves that form an outline and make the stroke color none and the fill color black.
6. Select **Object>Compound Path>Make** to make the multiple paths into a single object.
7. Save the file as an EPS file.
8. On the EPS Options dialog choose Illustrator 5/5.5 compatibility.

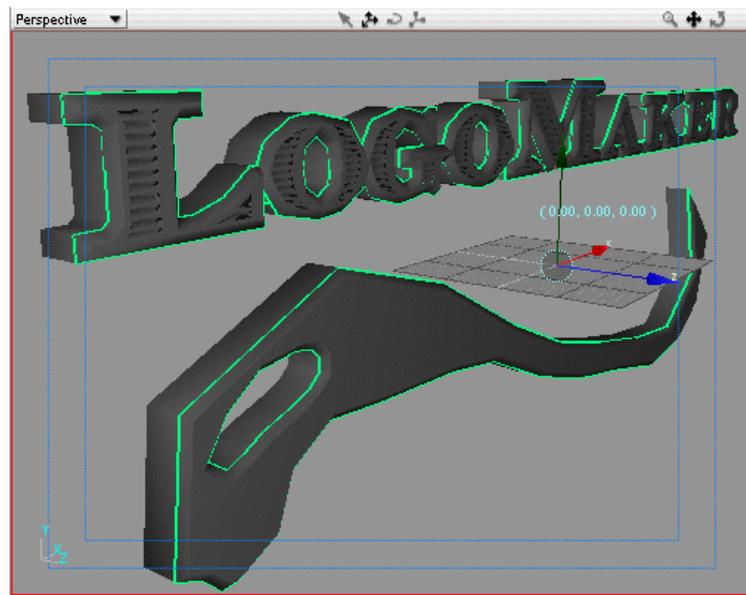
Importing the Text

1. In PiXELS 3D select **Special>LogoMaker>From Illustrator 5 File**.
2. Select the file you saved.

If some of the letters look inside-out, that usually means the path directions were incorrect in the original font. To fix this in Freehand, select the path and choose **Modify>Alter Path>Correct Direction** then save the logo and re-load it into PiXELS.

If a hand-drawn object is missing when you import the logo, that probably means one or more of the paths in the object were not closed.

Figure 11.1: Imported Logo



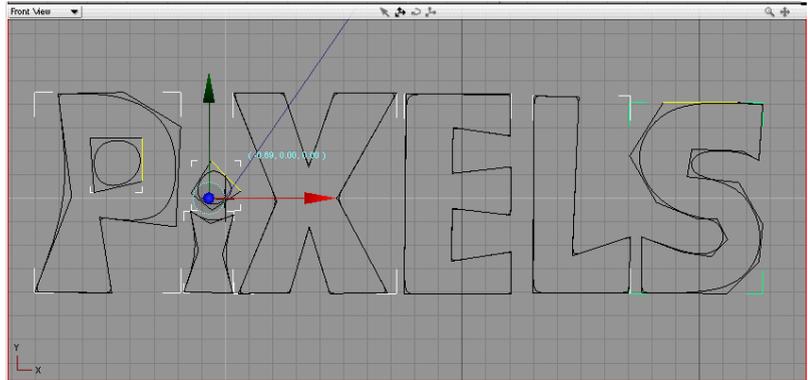
Drawing Splines in PIXELS 3D

1. Draw one or more closed (periodic) splines in any work view.

Note: The spline can be open (non-periodic), however, that produces letter shapes that are difficult to control.

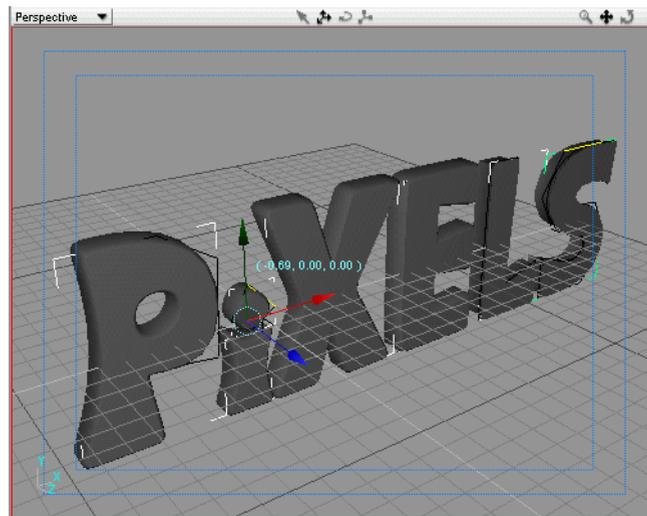
Note: If you accidentally draw the spline in the wrong direction, the resulting logo will look inside-out. To fix this, select the spline and choose **Edit NURBS>Invert>Invert U**.

Splines for Logo



2. With one or more splines selected, select **Special>LogoMaker>From Spline**. You will get a prompt to select additional splines. When you are finished selecting splines, press **Esc**.

Resulting Logo



Working with the Text

Object Attributes

The created logo object has several object attributes...

Transformation. These are similar to the Transformation attributes of other objects.

Render Options. The first three options, **Displacement Bounds**, **Shading Rate**, and **Shader**, are similar to the ones found in other objects. The **Shading Groups** option is unique to logo objects. By checking an **Override...Face** box and specifying a shader, you can set the various faces to use different shaders from the rest of the logo object.

Settings. The **Settings** options control the shape of the logo object.

The **Smooth...Face** options, when checked, create a smoothly-curved surface. When unchecked, the surfaces will be flat.



The **Breaking Angle** setting. This controls which edges will be hard edges and which edges will be smoothed when **Smooth Face** is selected. If the angle between faces exceeds the **Breaking Angle**, the edge between the faces will not be smoothed.

The **Show...Face** options determine whether a particular face is shown or hidden. For example, in the illustrations above, all of the faces are shown. You can create interesting effects by showing some faces and hiding others. To create an outline image, for example, check only **Show Extruding Face** and uncheck the others.



By combining **Shading Groups** with **Override Back Face**, **Show Front Face** and **Show Back Face**, you can get pseudo-shadow effects.



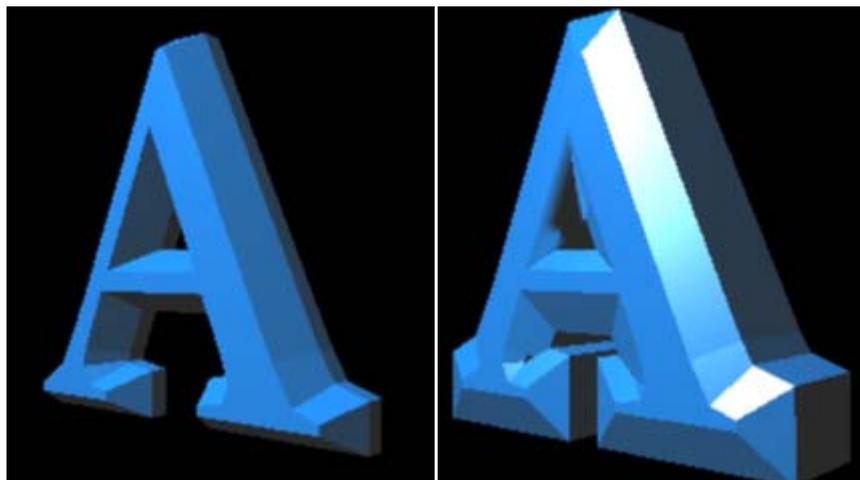
Flatness is the threshold angle for generating faces in the logo. That is, curves that exceed the flatness generate many small faces to fit the curve more tightly. Curves that stay within the threshold angle become one face on the resulting logo.

Extrude controls how wide the extruding face is. Setting this number to a higher value will make the back face farther away from the front face.

Bevel Size controls how wide the beveled face is; i.e., how far the bevel extends to the sides of the object.

Bevel Depth controls how deep the beveled face is; i.e., how far back the bevel extends from the front of the object.

The following images illustrate these attributes.



Extrude 1, Bevel Size 1,
Bevel Depth 2

Extrude 5, Bevel Size 2,
Bevel Depth 1

Bevel Inwards causes the bevel to extend inward, towards the center of the logo object, rather than outward, away from the object center. Depending on the size of the object, or the thick-

ness of one of its elements, you may need to change the **Bevel Size** to accommodate the inward bevel.



Bevel Inwards with
Bevel Size 1



Bevel Inwards with
Bevel Size 0.25

Display. These are similar to the **Display** attributes of other objects.

Tolerance. Controls how tightly the curve used to generate the logo fits the spline drawn in PiXELS.

Direction. This should be the same as the plane in which you drew the spline in PiXELS. For instance, if you drew the spline in the Front View (the XY plane), **Direction** should be set to XY.

AI File Path. This shows the complete file path of the Adobe Illustrator file used to create the logo object. If you make changes to the file, you can re-select the file to update your PiXELS 3D image.

Editable Format. You can make a logo into editable NURBS B-Splines or you can make the logo into an editable polygonal shape. Once you have created B-Splines from a logo, you can then make those B-Splines into a new logo. If you then animate the B-Splines with a modifier, you will animate the resulting logo.

Chapter 12. Polygonal Objects

Gallery of Objects

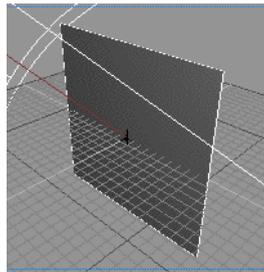
Polygonal objects are basic shapes that can be edited on a vertex by vertex basis or as whole objects by using Boolean tools. You can create a polygonal object by selecting an item from the **Polygons** menu or by clicking its icon in the **Favorites Bar**. Polygon object icons do not have a red frame like NURBS object icons.

2D

Square.

To create a square...

Select **Square** from the **Polygons>2D** menu or click the **Square** icon in the **Favorites Bar**.



Note: You can use the **Attribute Manager** to change the square's attribute values later.

Triangle.

To create a triangle...

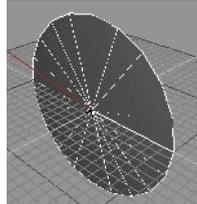
Select **Triangle** from the **Polygons>2D** menu or click the **Triangle** icon in the **Favorites Bar**.

Note: You can use the **Attribute Manager** to change the triangle's attribute values later.

Disk.

To create a disk...

Select **Disk** from the **Polygons>2D** menu or click the **Disk** icon in the **Favorites Bar**.



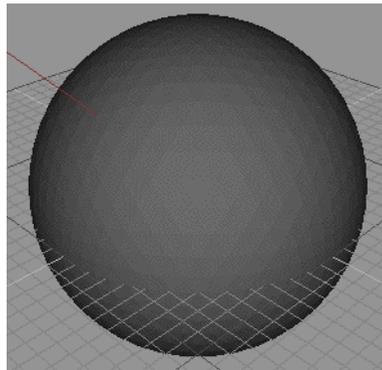
Note: You can use the Attribute Manager to change the disk's attribute values later

3D

Sphere.

To create a sphere...

Select **Sphere** from the **Polygons>3D** menu or click the **Sphere** icon in the **Favorites Bar**.

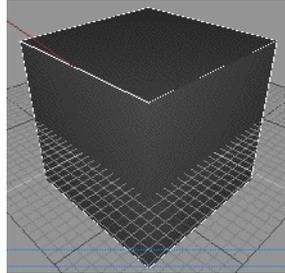


Note: You can use the Attribute Manager to change the sphere's attribute values later

Cube.

To create a cube...

Select **Cube** from the **Polygons>3D** menu or click the **Cube** icon in the **Favorites Bar**.

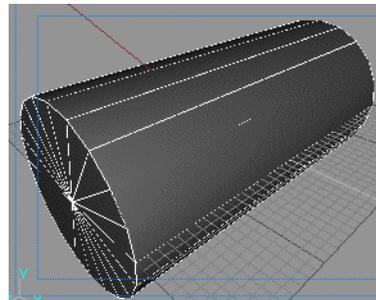


Note: You can use the Attribute Manager to change the cube's attribute values later.

Cylinder.

To create a cylinder...

Select **Cylinder** from the **Polygons>3D** menu or click the **Square** icon in the **Favorites Bar**.

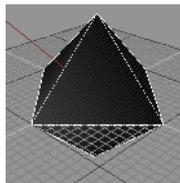


Note: You can use the Attribute Manager to change the square's attribute values later.

Octahedron.

To create an octahedron...

Select **Octahedron** from the **Polygons>3D** menu or click the **Octahedron** icon in the **Favorites Bar**.

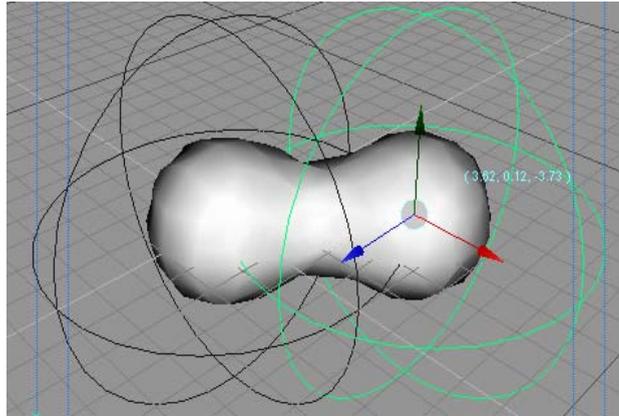


Note: You can use the **Attribute Manager** to change the octahedron's attribute values later.

Chapter 13. Metaballs

Metaballs, also called blobbies, are systems that dynamically determine surfaces. These systems look something like soap bubbles or blobs of clay.

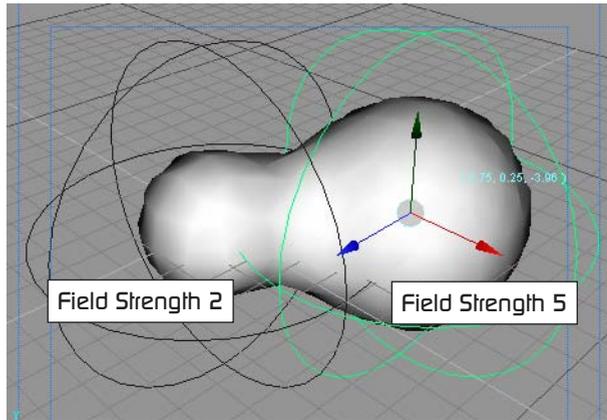
Figure 13.1: Two Blobbies in a Metaball System



[\[Click the picture to view the animation\]](#)

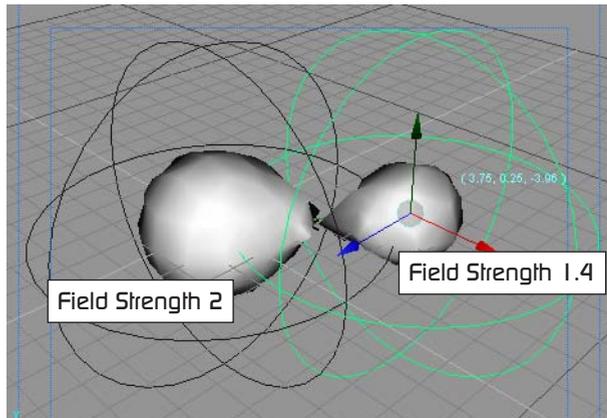
When blobbies in the same system are close together their surfaces join like beads of water on a waxed surface. Under each blobby object in the Attribute Manager there is an attribute called **Field Strength**. This controls the surface around that blobby. When the **Field Strength** is large, the blobby is big.

Figure 1 3.2: Large (5) Field Strength



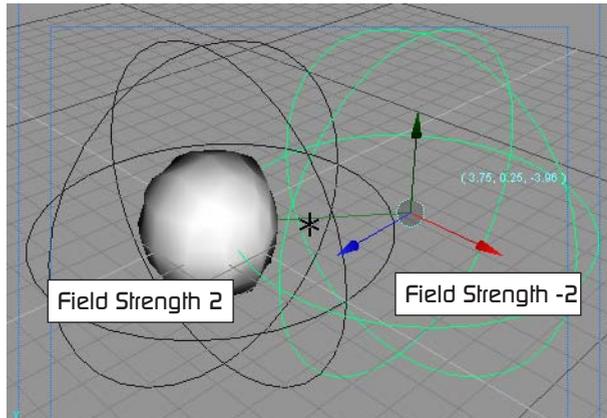
When the Field Strength is small, the blobby is also small.

Figure 1 3.3: Small (1.4) Field Strength



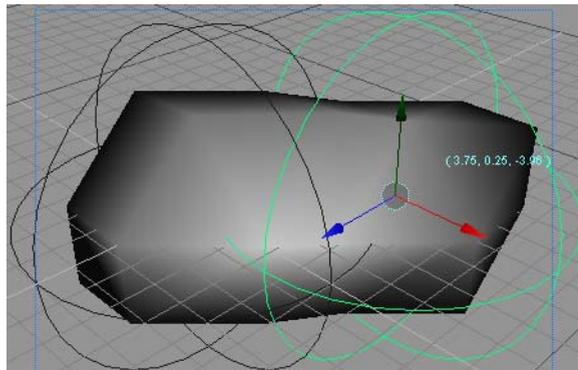
You can make a blobby repel other blobbies by making the Field Strength negative.

Figure 13.4: Negative (-2) Field Strength



By default, the on-screen appearance of the blobbies looks very coarse. This allows the blobby surfaces to be calculated quickly. To get a more accurate on-screen picture of the system reduce the **Subdivision Level (Editor)** value. You can find this value under the blobby system in the Attribute Manager.

Figure 13.5: Default Subdivision Level (Editor)



Chapter 14. Shading

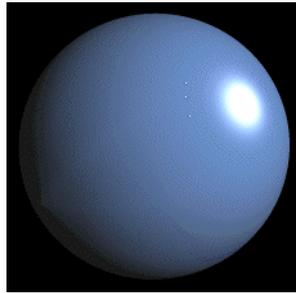
Shaders vs. Materials

Many 3D programs use a single method to represent the interaction between light and objects. In these programs you use what are typically called *materials* to give objects a particular color, bumpiness, or general appearance. A material describes the surface of an object. The 3D program then uses its built-in light interaction procedures to cast light on the material covered object and produce an image.

A *shader* is a little bit like a material, only more flexible. While a material describes surface attributes, a shader describes how the interaction between light and the surface will be represented.

For example, many programs use a method of interaction between light and surfaces called Phong Illumination.

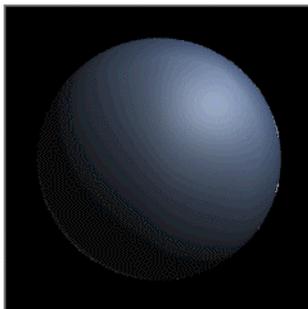
Figure 14.1: Phong Illumination



This particular illumination method makes objects look a little like plastic. One of the characteristics of Phong illumination is what is called its *specular function*. A specular function describes how light produces the bright spot, or *specular highlight* on an object.

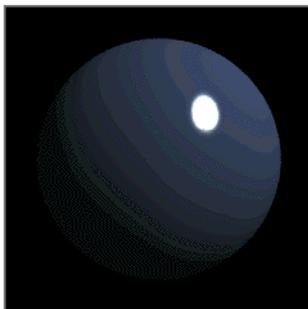
PiXELS shaders have a whole assortment of specular functions for you to choose from. For instance you could choose Blinn for a more subtle highlight.

Figure 14.2: Blinn Highlight



You could use Aniso for a sharply defined highlight that can be elongated.

Figure 14.3: Anisotropic Highlight



You could also use any of the other highlight functions and any other general shading functions for more complex light interactions.

Figure 14.4: Glossy Highlight with Velvet

**Subsurface Scatter**

Many surfaces do not simply reflect light. Instead, some of the light passes into a translucent area at the surface, bounces around in this translucent area, then is diffusely reflected by the opaque surface beneath the translucent layer. Porcelain, human skin, enamel, and velvet all share this characteristic. Use the `skin` or `velvet` nodes to simulate this effect.

Volumetrics

Usually shading is calculated at the surface of an object. This works well in most situations. However, when you want some kind of fog effect, you want the amount of fog to increase as the distance between the object and the camera increases. The atmosphere shader accomplishes this through a volumetric technique called ray marching.

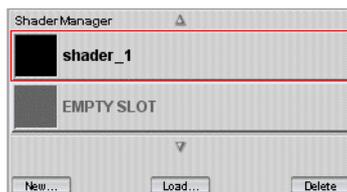
Shading Interface

There are three widgets or panes to the shading interface. `ShaderManager` lets you organize and select shaders. `TextureManager` lets you load bitmaps for use in your shaders. `ShaderMaker` lets you construct new shaders and modify existing shaders.

ShaderManager

`ShaderManager` lists all of the shaders in your scene.

Figure 14.5: ShaderManager



To create a new shader,

Click **New** in the **ShaderManager**. This opens a new shader in **ShaderMaker** and creates an entry for the new shader in the **ShaderManager** list.

To edit an existing shader,

Scroll up or down the list by clicking on the arrows and select the shader. This opens the shader in **ShaderMaker**.

Shaders get saved when you save your scene; however, sometimes you want to save a shader without saving the rest of your scene.

To save a shader,

Choose **Save** from the **File** menu in **ShaderMaker** (the menu is in **ShaderMaker**, not **ShaderManager**).

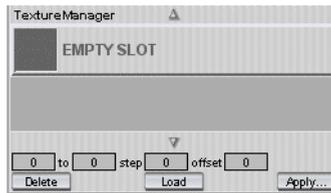
To load a shader from another scene,

Click on **Load** in **ShaderManager** and select the shader you want to load.

TextureManager

TextureManager lists all of the image maps in your scene.

Figure 14.6: TextureManager



To load a new image map,

Click **Load** and select your image in the resulting file browser.

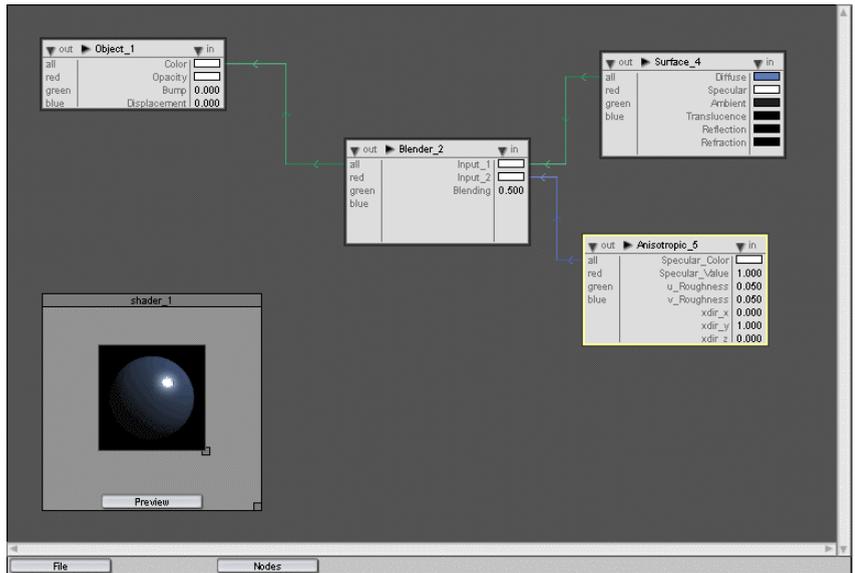
To apply an image map to a shader...

1. Select image map from the **Nodes** menu (in **ShaderMaker**). This creates a new image map node.
2. Click the **image source** field in the new node. This opens a menu that lists all the images in **TextureManager**.
3. Choose the image map you want to use.

ShaderMaker

ShaderMaker lets you graphically construct shaders.

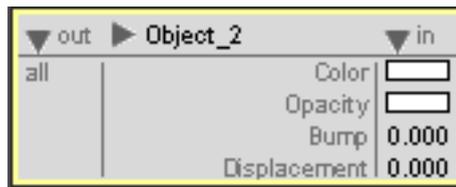
Figure I 4.7: ShaderMaker



Object Node

Shaders are constructed from nodes, each of which has a number of input parameters and an output. When a new shader is created (by clicking **New** in **ShaderManager**), one node called the *object node* is created.

Figure I 4.8: Object Node



This node is what actually shades the object. Any nodes that can trace a connection to one of the four input parameters on the **Object** node affect the shader appearance.

object input parameters

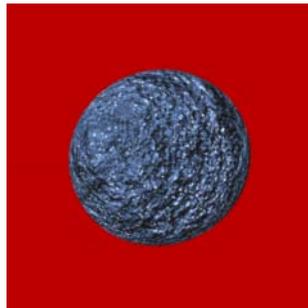
Color. The color of the object. For a simple, uniform, unshaded color, click on the color box. This opens the operating

system **Color Picker** where you can choose a color. Typically you want to connect other nodes to the **Color** input parameter.

Opacity. The way that the object blocks light. If you assign a color with a grey **Value** of less than 100%, light will appear to shine through the object. That light will appear to have the **Hue** and **Saturation** assigned to the **Opacity** input. Like other input parameters on the **Object** node, you typically want to connect other nodes to the **Opacity** input.

Bump. This is a way that 3D programs imitate surface roughness. **Bump** alters what are called the surface normals to make the surface look bumpy without actually displacing the surface.

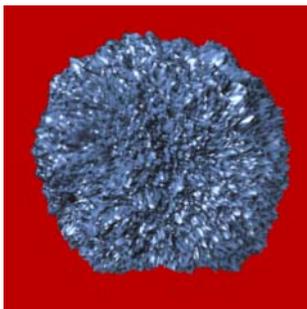
Figure 14.9: Surface with fBm Bump



In Figure 14.9, notice how the edge of the sphere looks smooth even though the surface looks bumpy. The **Bump** value multiplies the input values received through a connector. This means an object with a **Bump** of 0 will always look smooth, regardless of what is connected to the input.

Displacement. While **Bump** imitates a rough surface, **Displacement** actually creates a rough surface.

Figure 14.10: Surface with fBm Displacement



In Figure 14.10, notice how both the edge and the surface of the sphere look bumpy. Because of the way Tempest works it can render **Displacement** shaders nearly as fast as it can render **Bump** shaders. However, with larger displacements you will sometimes get seams on the surface.

To get rid of displacement seams,

Increase the **Displacement Bounds** in the **Render Options** for your object (this can be found under your object in the **Attribute Manager**).

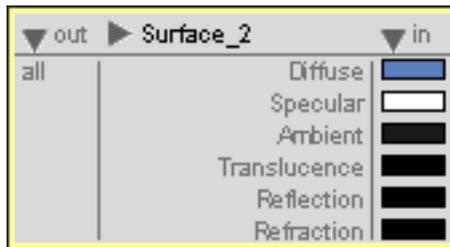
To get rid of seams in the **ShaderMaker Preview**, increase the **Displacement Bounds** under **ShaderMaker Options** in the **Attribute Manager Preferences**.

Note: Increasing the **Displacement Bounds** increases memory usage and render time, so for complex scenes keep **Displacement Bounds** as low as you can.

Nodes

A node is the basic unit of a shader. A node can be thought of as a single type of light-object interaction. Each node has multiple input parameters and one output channel.

Figure 14.11: Basic Surface Node



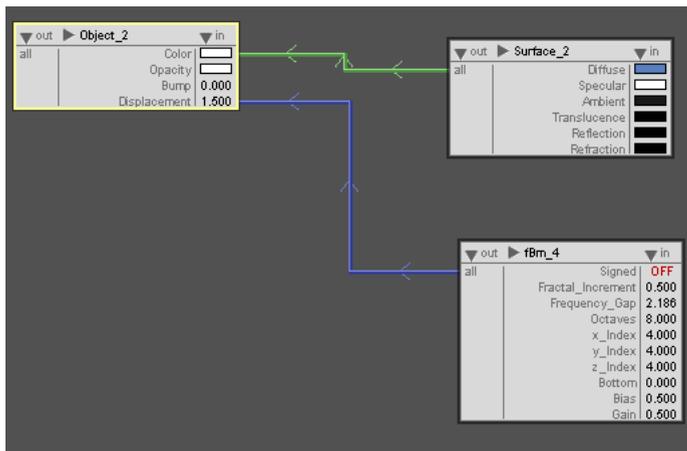
To create a new node,

Select the node type from the **Nodes** menu or from the **ShaderMaker** contextual menu (Control-Click in **ShaderMaker** to open the contextual menu).

To connect a node...

1. Click and drag from the output channel (under **out**), to an input parameter (under **in**) of another node. This draws a line from the output of one node to the input of the other.
2. If the input parameter is numeric, it should be set to a non-zero number. If the input is a color, it should be set it to a non-black color (typically white). The input received through the connection is multiplied by that number or color (See [Color Multiplication](#) on page 174).

Figure 14.12: Two Nodes Connected to the Object Node



(Click the picture to view the animation)

To show or hide the output, preview, or input,

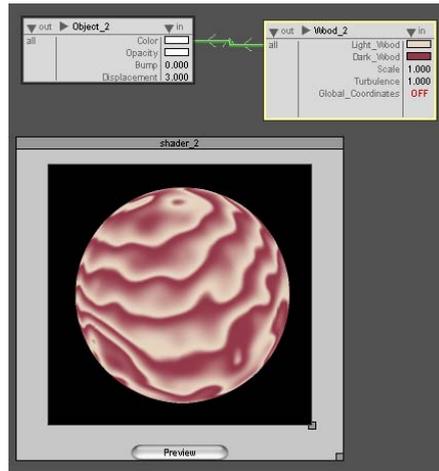
Click on the arrow next the out, the shader name, or in respectively.

Color Multiplication

When you plug a line into an input field with a color, the color sent to the node is multiplied by the color in the input field. Color multiplication is like putting a colored lens filter on your shader. If you are not familiar with the concept, color multiplication may be a little tricky to grasp without a few examples.

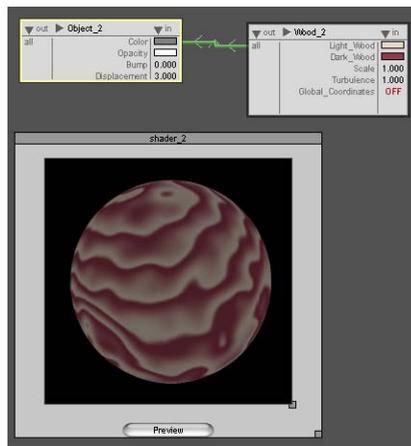
Multiplication by white is like normal multiplication by 1 (or like a clear lens filter). That is, the result is the same as the input.

Figure 14.13: Wood Multiplied by White



Multiplication by black is like multiplication by 0 (or a lens cap). Multiplication by grey, is like multiplication by a number between 0 and 1 (or like a dark lens filter).

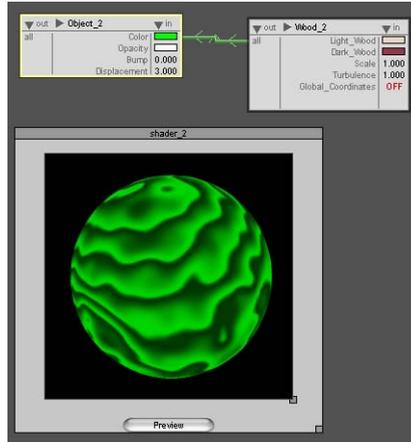
Figure 14.14: Wood Multiplied by Grey



Multiplication by a primary color (red, green, or blue), multiplies that color channel by 1 and multiplies the other primary

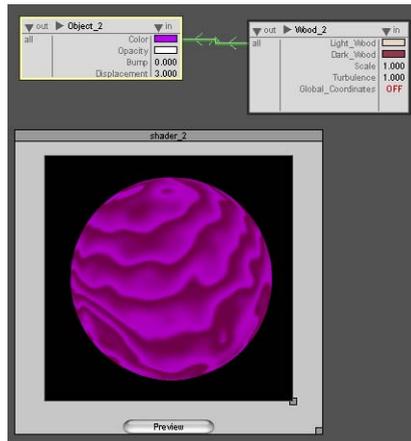
channels by 0. In essence, it filters out the other two primary colors.

Figure 14.15: Wood Multiplied by Green



Multiplication by other colors filter out the three primary colors in varying degrees.

Figure 14.16: Wood Multiplied by Lavender



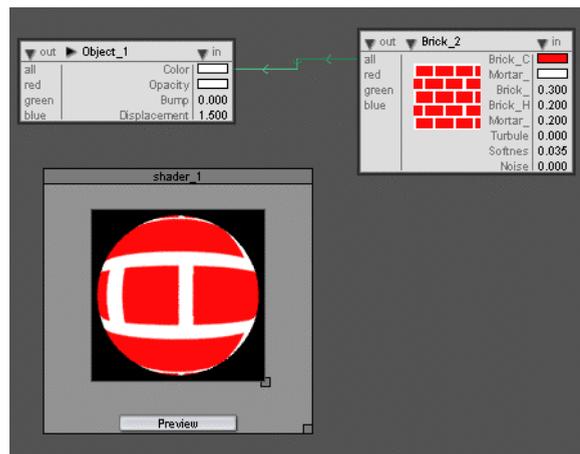
Connecting Nodes

Nodes interact with the lighting and object shape to produce an image across the object. That image is then transmitted to

input parameters of the connected nodes. Those connected nodes then interact with the lighting, object shape, and the variable input received through the connection. In this way each node adds its contribution to the overall shader.

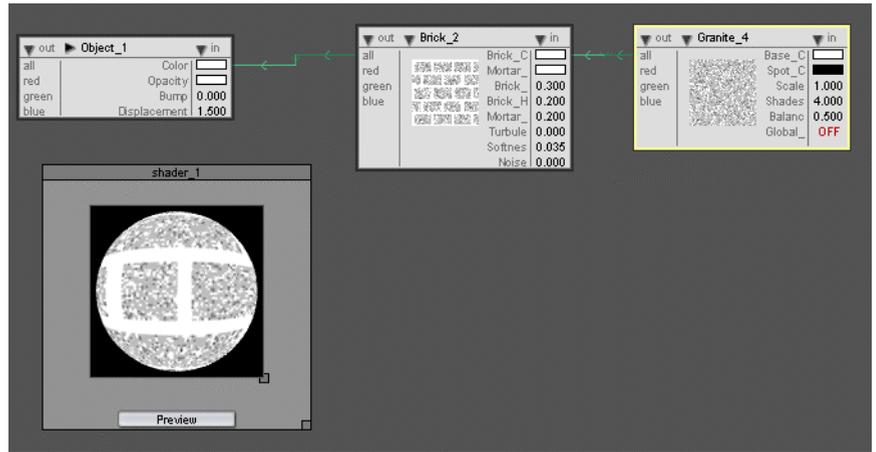
For example, a **Brick** node produces an unshadowed surface colored with a brick pattern.

Figure 14.17: One Node



If you connect a **Granite** node to the **Brick Color** input, the combined nodes produce an unshadowed surface colored with granite textured bricks.

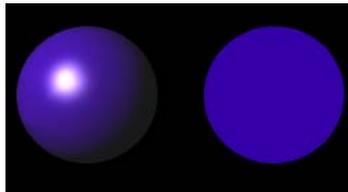
Figure 1 4.1 8: Two Nodes



Surface Nodes

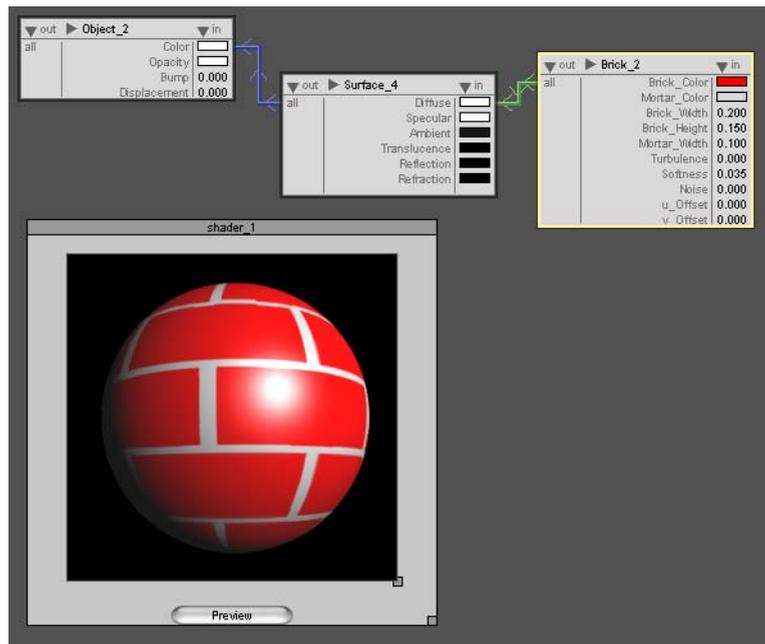
If you want shadowing on your surface you need to add one of the basic surface nodes. These are *surface*, *sm3_surface*, *skin*, *diffuse*, *clay*, and *velvet*. Without one of these nodes your object will appear luminous and flat.

Figure 1 4.1 9: Purple Sphere with and without Surface



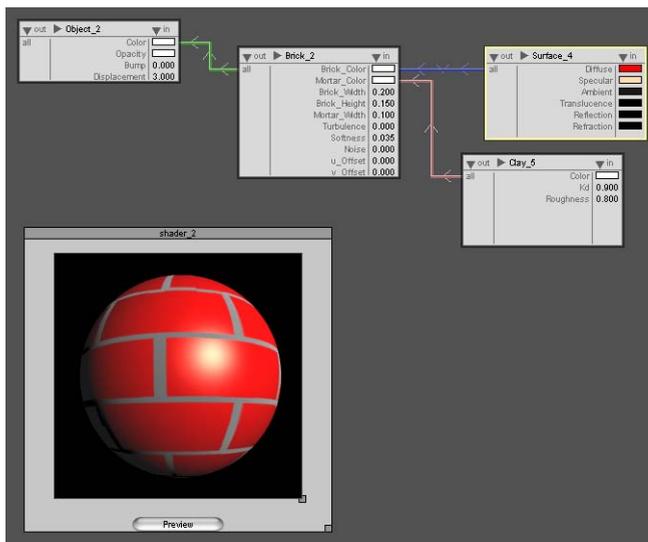
If you want to add a surface to a pattern (brick, for instance), you can place one surface node between the object node and the last node before the object node.

Figure 14.20: Brick with a Single Surface



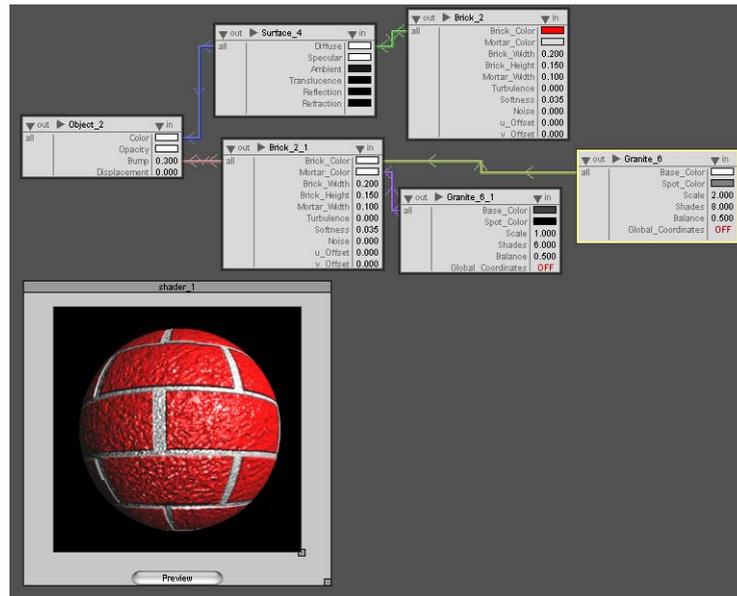
You can also add a surface node input to each color in the pattern. This way the brick can have different surface properties from the mortar (notice the mortar looks dull and the brick looks shiny).

Figure 1 4.2 1: Brick with a Surface and a Clay Node Added



When you have a patterned or textured object with a basic surface, the object often looks best if you add bump or displacement with a similar pattern.

Figure 14.22: Bump Added



Editing Inputs

There are several different kinds of inputs on a node.

Numeric fields are straightforward to edit, you just select the number and type a new one.

For color fields you click on the color swatch (the colored rectangle on the node input). That opens the standard Macintosh color selector. Select the color you want and click OK. To numerically edit the red, green, and blue values, Option-click on the color swatch.

Selection fields appear in red. To edit those fields, click on the field to open a pop-up menu, then select the item you want from the menu.

Vector fields appear as a red V. The vector is composed of 3 numbers. Click on the V to see the individual numbers. To edit any of the numbers, click on the number and type a new one.

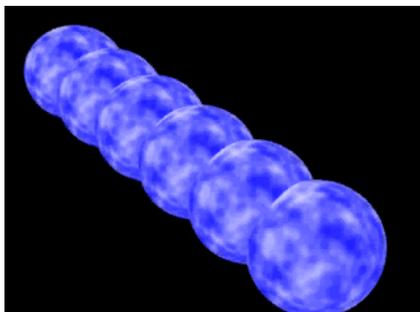
Ka, Kd, and Ks

Some nodes have fields for values labeled K_a , K_d , and K_s . These are abbreviations for ambient reflectivity, or glow (K_a), diffuse reflectivity, or surface brightness (K_d), and specular reflectivity or highlighting (K_s).

Global Coordinates

Shaders with solid patterns (any of the 3D textures) have a **Global Coordinate** field that can be **ON** or **OFF**. When **OFF**, the texture sticks to the object surface as you move the object around. This is normally what you would expect.

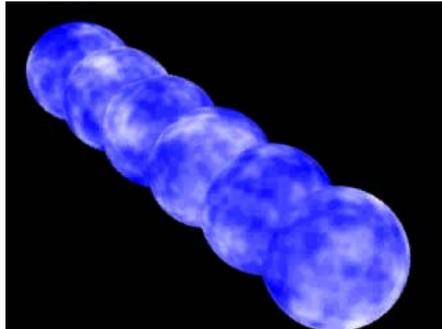
Figure 14.23: Clouds with Global Coordinates OFF



(Click the picture to view the animation)

When **ON**, the texture sticks to the scene, so when you move the object, it looks like it is moving through the texture.

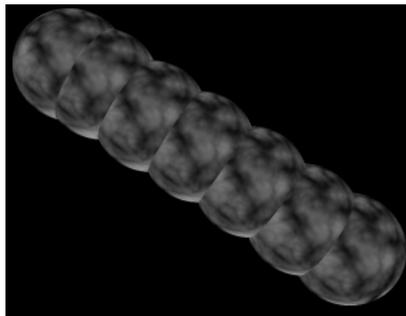
Figure 14.24: Clouds with Global Coordinates ON



(Click the picture to view the animation)

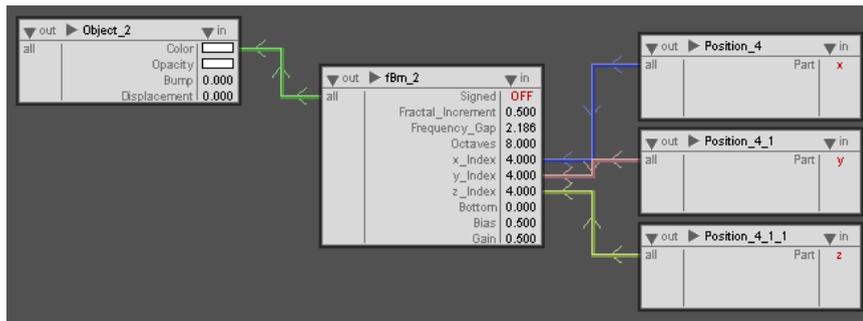
The noise functions (fBm, Fractal Sum, and Turbulence) always use local coordinates.

Figure 14.25: Turbulence Connected to Color



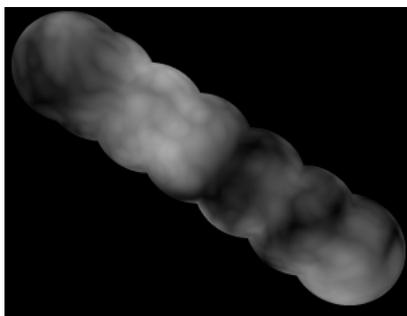
If you want a noise function to become global, attach **P** nodes (found under **point nodes > globals** on the **Nodes** menu). to the **x**, **y**, and **z** inputs. You will need one **Position** node for each input. The one connected to **x** should have **Part** set to **x**. The one connected to **y** should have **Part** set to **y**. The one connected to **z** should have **Part** set to **z**.

Figure I 4.26: Shader for Global Turbulence



This produces turbulence that behaves like a global solid.

Figure I 4.27: Global Turbulence

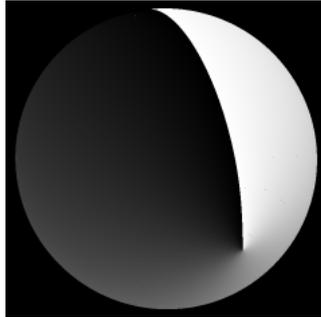


2D Surface Textures

If you want to attach a noise or pattern to the 2D surface of an object, rather than to the object volume, use **s** and **t** nodes.

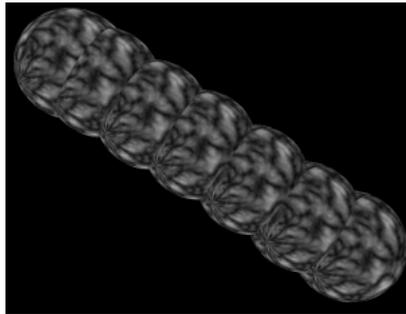
s and **t** are texture surface coordinates corresponding to **u** and **v** respectively. This means, as the **u** value on a NURBS surface increases, so does the **s** value. As the **v** value increases, so does the **t** value. **s** and **t** range in value from 0 to 1, so if you connect one of these nodes to color, the object color is a gradation from black to white.

Figure 14.28: s Connected to Color



If you connect s to the x and z inputs of a noise function and connect t to the y input, the noise will stick to the object surface like a decal wrapped around the object.

Figure 14.29: s and t Connected to Turbulence



Once the noise is tied to the surface coordinates (s and t) the pattern is duplicated on each sphere regardless of its position. However, unlike the default turbulence, you also get pinching and stretching as you distort the NURBS surface (notice the pinching at the sphere poles).

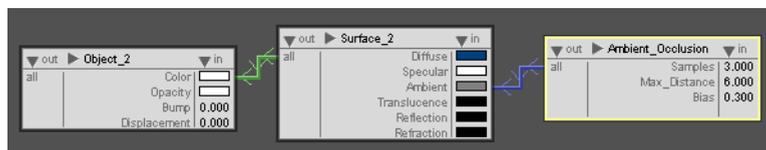
Gallery of Nodes

Ambient Occlusion

Allows ambient light to add illumination to an object without washing out the shadows.

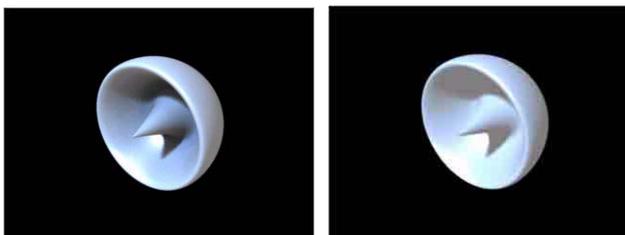
The ambient node works by sending out rays from each point on the surface, then illuminating the surface according to how many rays intersect objects. So, if the surface faces the ground or is close to the other object there will be less ambient illumination than if the surface faces the sky.

Figure 1 4.30: Raytraced Functions>Ambient Occlusion



Note: The ShaderMaker preview does not completely represent the effects of raytraced nodes, including the Ambient Occlusion node.

Figure 1 4.3 1 : Ambient Occlusion Example



Ambient Occlusion On

Ambient Occlusion Off

Samples. The average number of rays emitted from each point. Increasing the **Samples** value increases the realism of the shadows, but also increases processing time.

Max Distance. The maximum length of the rays used. Higher values create shadows from more distant objects.

Bias. The ray bias. This adds a jittered offset value to the ray. On flat surfaces and unsmoothed polygons, you can set this value very low (0.0001). On concave NURBS surfaces you may need to make Bias as high as 0.3 (or even higher on rare occasions).

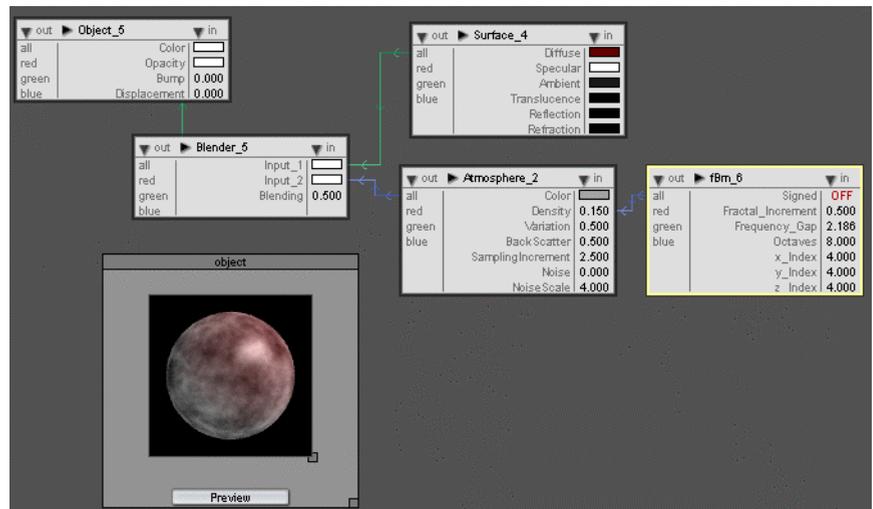
Atmosphere

Atmosphere is a little different than most nodes. It creates a volumetric effect between the object to which it applies and the camera.

To use an Atmosphere node...

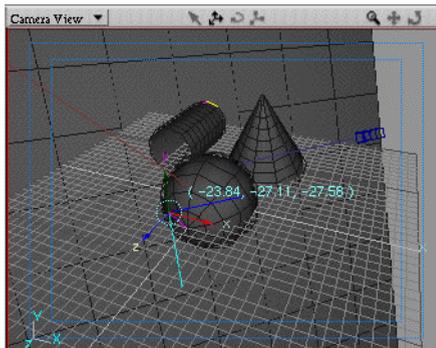
1. Create shaders for your objects that mix in the atmosphere node.

Figure 14.32: Basics > Atmosphere (Object)



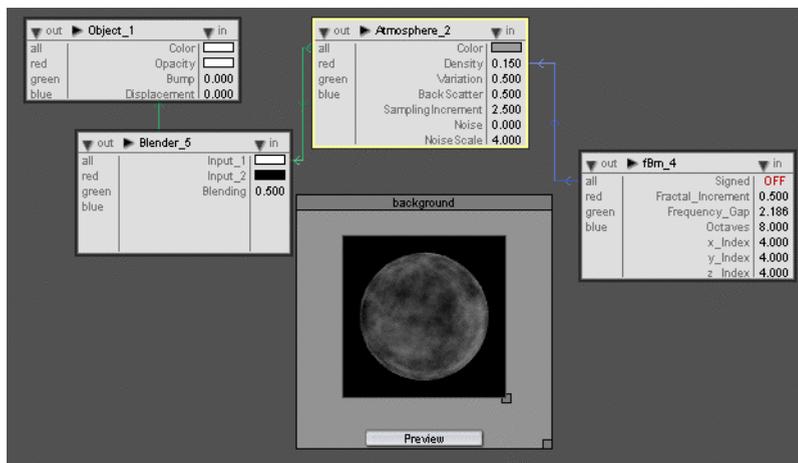
2. Create a backdrop object to fill the background with atmosphere.

Scene with Mesh as Backdrop



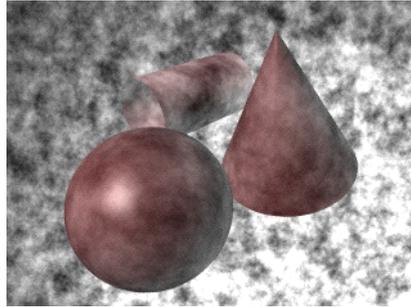
3. Create a shader that mixes in the atmosphere node for your backdrop.

Basics>Atmosphere (Background)



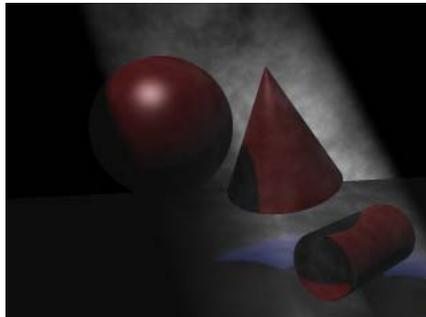
4. Make sure an object or the backdrop covers everything visible from the camera and render your scene.

Figure 14.33: Scene Using Atmosphere



If you shine a spotlight through a scene with atmosphere, you can get great volumetric lighting effects.

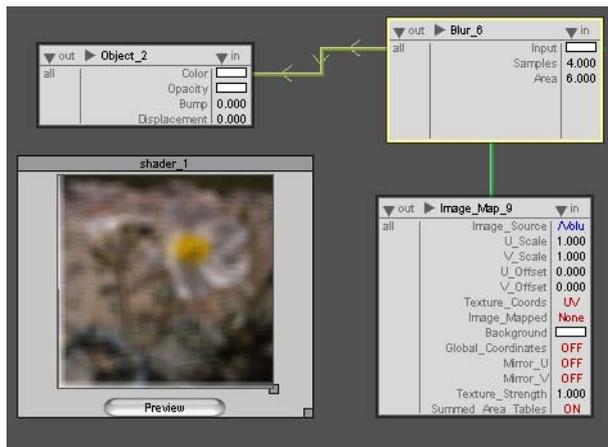
Figure 14.34: Atmosphere with Narrow Spot



Note: If you use shadow mapping with atmospheric lighting, make sure the shadows are cast onto an object with a basic surface, otherwise the fog within object shadows will be too bright.

Blur

Figure 1 4.35: Utility Functions>Blur



Here is the original image, which is speci ed under Image Source in the Image Map node:



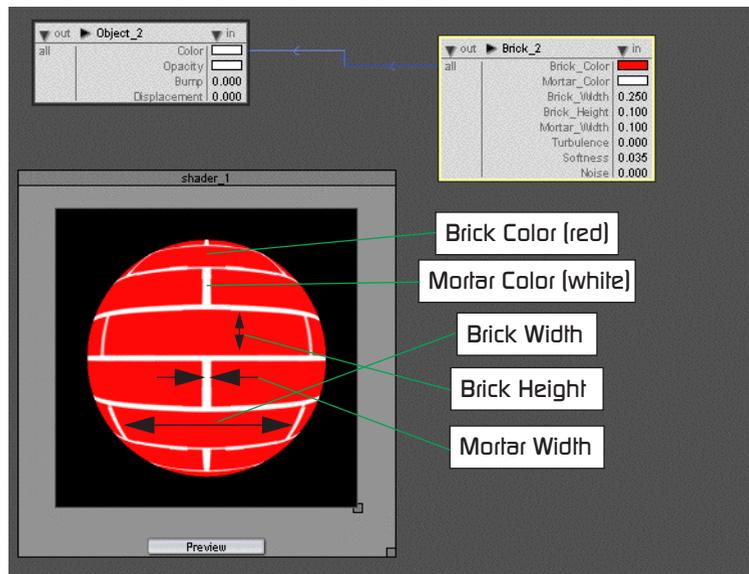
Input. The source image to be blurred. Drag **all** from the out column of the Image Map node to the Input of the Blur node.

Samples. Increase the **Samples** value to increase the overall image quality and smooth out roughness.

Area. Increase the **Area** value to make the image more blurry.

Brick

Figure 14.36: 2D Textures > Brick



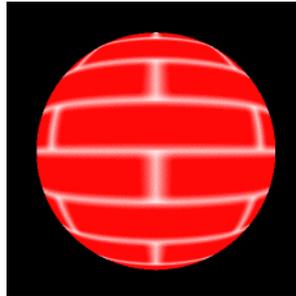
Turbulence. Adds randomness to the path of the mortar lines.

Figure 14.37: Brick with Turbulence 0.1



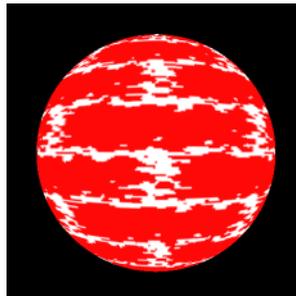
Softness. Adds fuzziness to the boundary between the mortar and the brick.

Figure 1 4.38: Brick with Softness 3



Noise. Adds randomness to the mortar thickness.

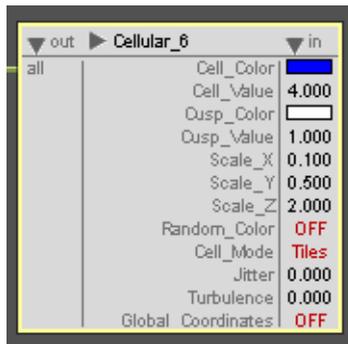
Figure 1 4.39: Brick with Noise 0.5



Cellular

Creates mosaic-like or cubic tiles.

Figure 14.40: 3D Textures>Cellular



Cell Color. The color of the cells.

Cell Value. With **Cusp Value**, determines the dominance of the cells vs. the cusps.

Cusp Color. The color of the borders between the cells.

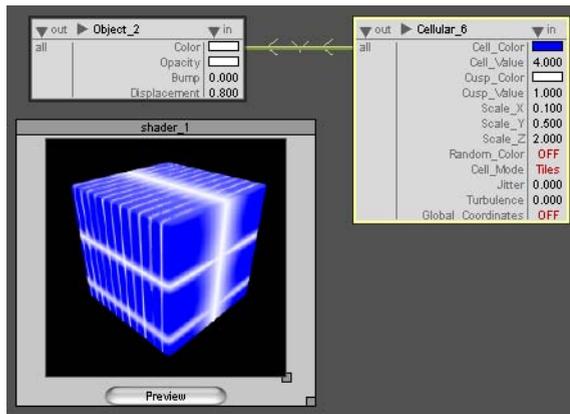
Cusp Value. With **Cell Value**, determines the dominance of the cells vs. the cusps.

Scale X. Changes the size of the tiles in the x direction.

Scale Y. Changes the size of the tiles in the y direction.

Scale Z. Changes the size of the tiles in the z direction.

Figure 14.41: Tiles with Scale Values Set and No Jitter

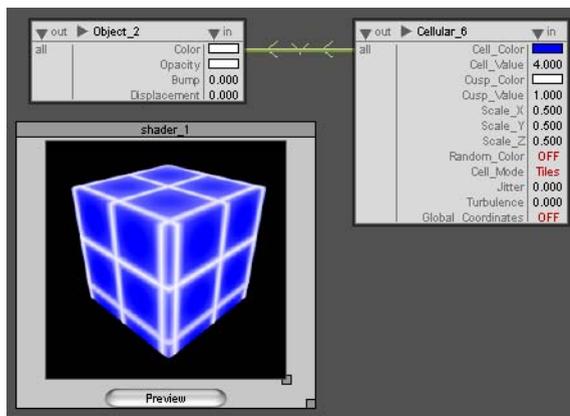


Random Color. If Random Color is turned on, the color of each cell will be determined randomly. See Figure 14.46 on page 196 for an example.

Cell Mode. Determines the type and shape of the cells.

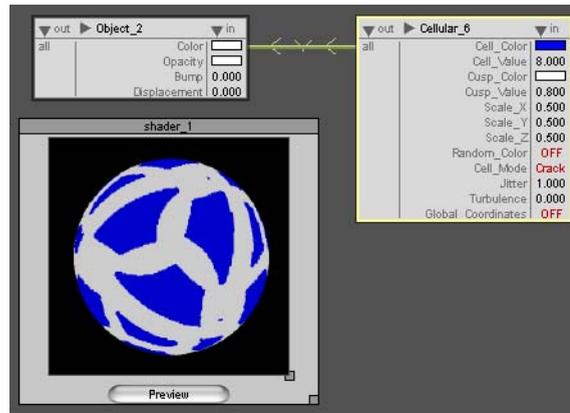
Tiles. Creates three-dimensional tiles. Increase the Cell Value to make the tiles more prominent.

Figure 14.42: Cell Mode Set to Tiles



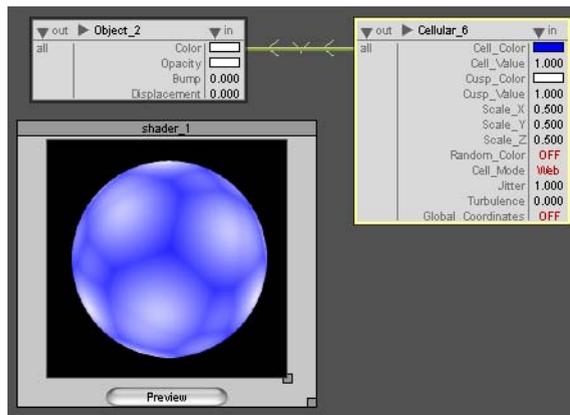
Cracks. Emphasizes the cracks between the cells. Set Cell Value fairly high.

Figure 14.43: Cell Mode Set To Cracks



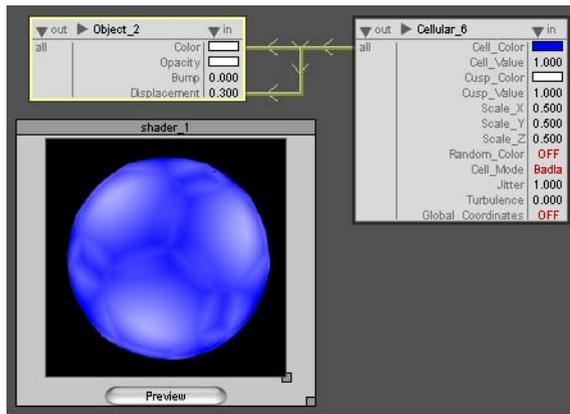
Web. Creates a more spidery web, with the Cell Color forming the strands of the web. Set the Cell Value fairly low.

Figure 14.44: Cell Mode Set To Web



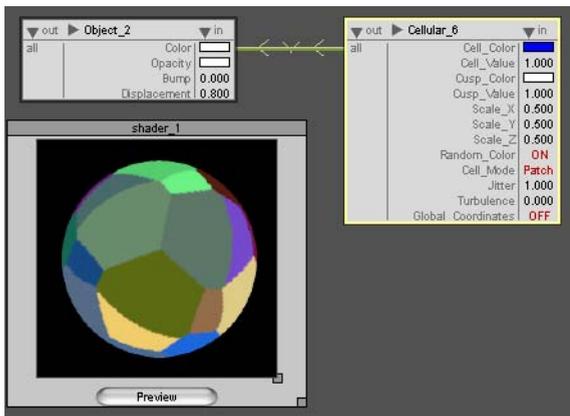
Badlands. Like Web, but adds an additional dimension.

Figure 1 4.45: Cell Mode Set To Badlands



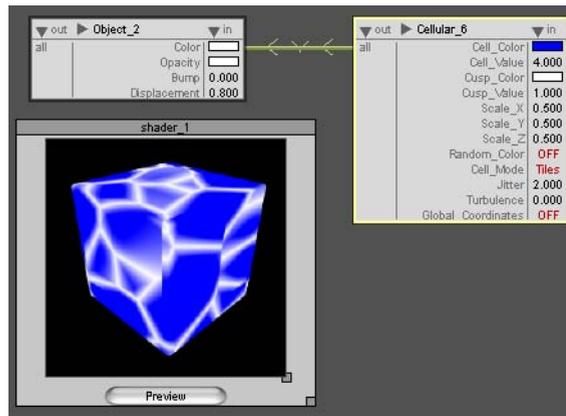
Patchwork. Creates a crazy-quilt style surface when Random Color is turned on.

Figure 1 4.46: Cell Mode Set to Patchwork



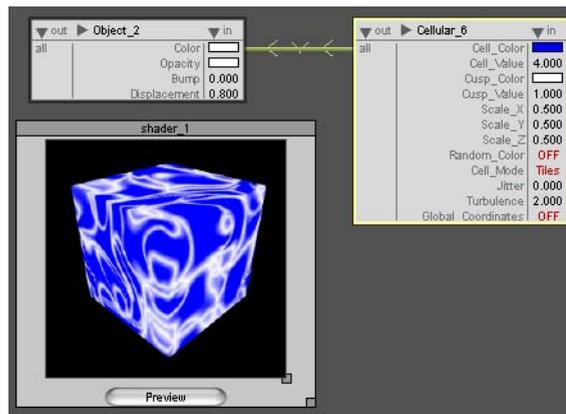
Jitter. Makes the cells size more irregular.

Figure 14.47: Tiles with Jitter



Turbulence. Makes the boundary between cells more irregular.

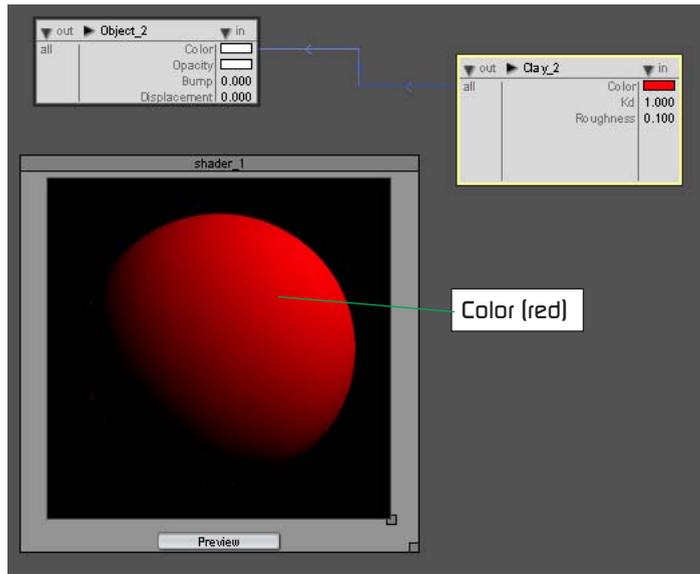
Figure 14.48: Tiles with Turbulence



Global Coordinates. Causes the object to use the global coordinates when turned on.

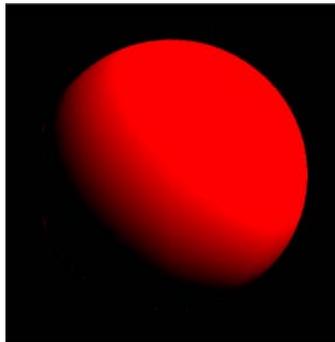
Clay

Figure 1 4.49: Diffuse Functions>Clay



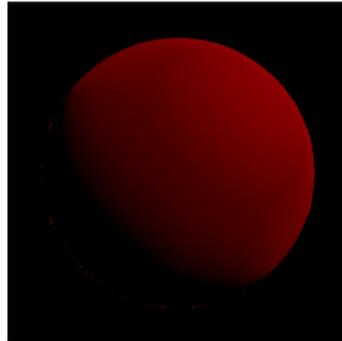
Kd. Diffuse re ectivity. Higher values increase the brightness closest to the light.

Figure 1 4.50: Clay with Kd 2



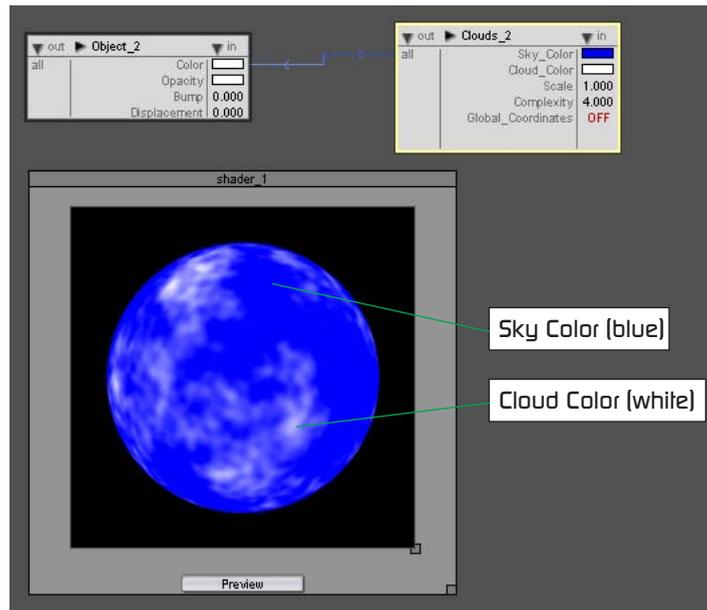
Roughness. A rougher surface gives a more velvet-like appearance.

Figure 14.51: Clay with Roughness 2



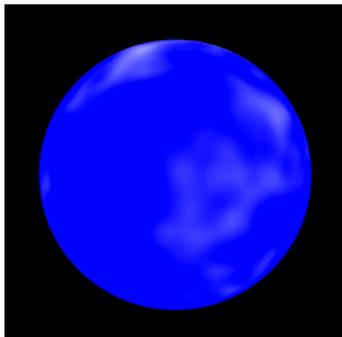
Clouds

Figure 14.52: 3D Textures>Clouds



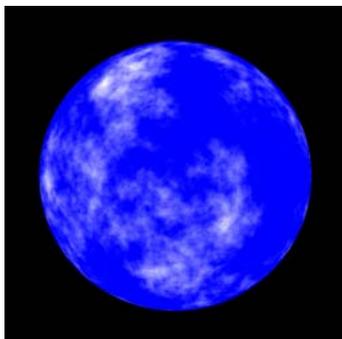
Scale. The size of the clouds.

Figure 14.53: Clouds with Scale 2.5



Complexity. The amount of detail in the cloud pattern.

Figure 14.54: Clouds with Complexity 8



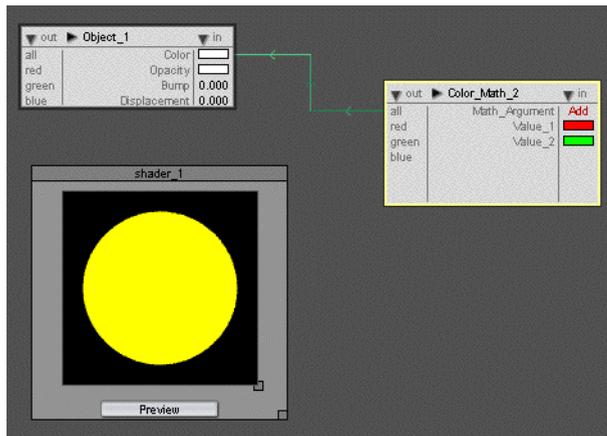
Comp (Component)

Returns the red (0), green (1), or blue (2) component of the **Color**, and the x (0), y (1), or z (2) component of the **Point**.

Color Math

Performs the specified math function on the red, green, and blue values of the two input colors.

Figure 14.55: Color Nodes > Color Math



Math Argument. The mathematical function.

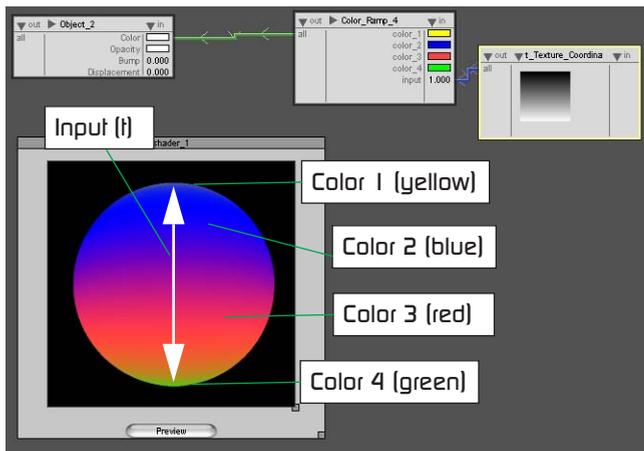
Add. Adding color is like mixing light.

Multiply. Multiplying color is like projecting your shader through a colored filter. (For a detailed explanation of color multiplication, see [Color Multiplication](#) on page 174.)

The other math functions are more difficult to characterize in an intuitively understandable manner. If you want to see what a particular math function does to color, try experimenting with it for a bit. For more detail about some math functions see [Math Functions](#) on page 216.

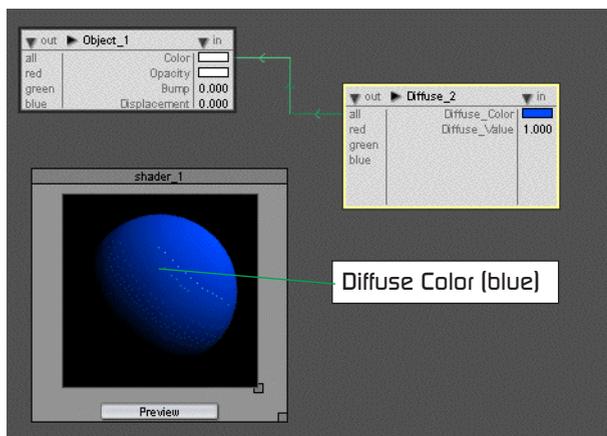
Color Ramp

Figure 1 4.56: Color Nodes>Color Ramp



Diffuse

Figure 1 4.57: Diffuse Functions>Diffuse

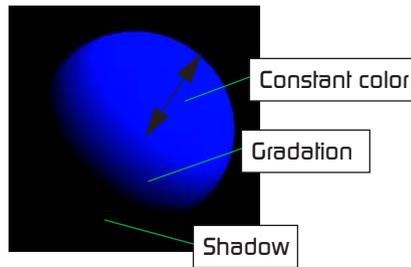


Diffuse Value. The brightness or darkness of the surface. For a darker surface, make the value a number between 0 and 1.

The **Diffuse Color** is the brightest the surface can get, so if you specify a number greater than 1 for **Diffuse Value**, the surface will be black in the shadow, there will be a gradation from

black to the Diffuse Color, then the rest of the object will be evenly colored with the Diffuse Color.

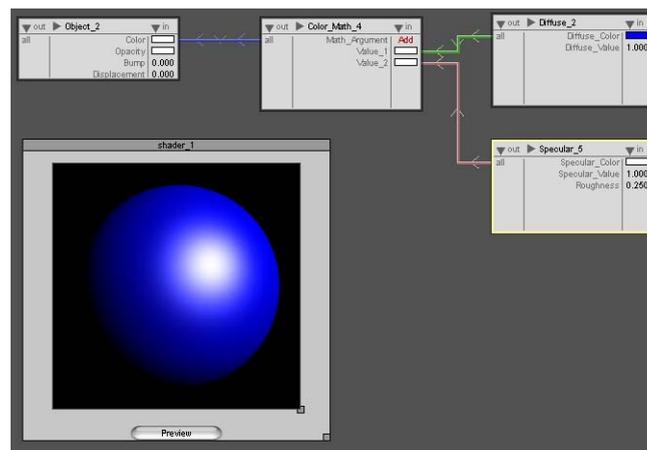
Figure 14.58: Diffuse Value 2



To add a highlight to a Diffuse surface...

1. Create a Color Math node set to Add.
2. Connect a Specular Functions node to one input on the Color Math node (this creates the highlight).
3. Connect the Diffuse node to the other input on the Color Math node.
4. Connect the Color Math node to the Color input on the Object node.

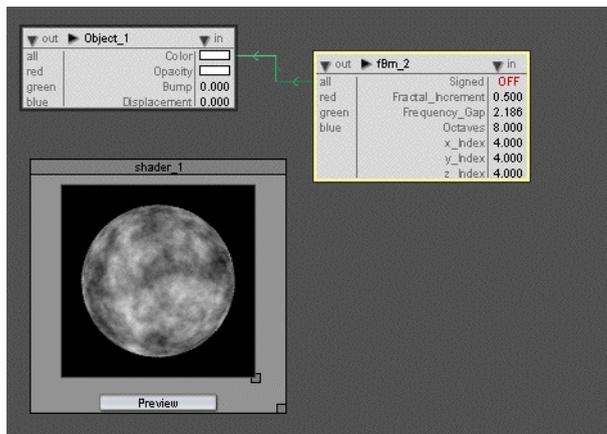
Diffuse Node with a Specular Highlight Added



fBm

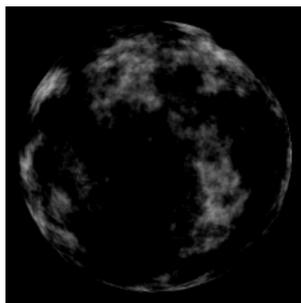
Fractal Brownian motion noise function. This is a particular type of well defined randomness.

Figure 14.59: Float Nodes > fBm



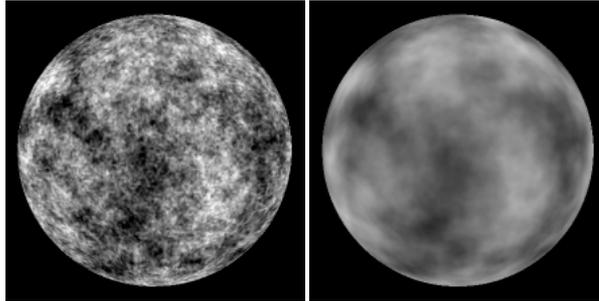
Signed. Allows values to be negative. (If a negative value is connected directly to Color, it is black, so the result of turning this on when connected to Color would be larger areas of black.) When signed is **OFF**, the output is the absolute value of the noise.

Figure 14.60: Signed fBm



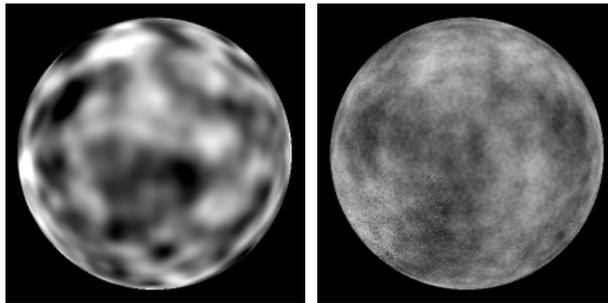
Fractal Increment. A lower fractal increment gives you more crisp detail. A higher fractal increment gives you a more blurred effect.

Figure 14.61: fBm with Fractal Increment of 0.1 and 1



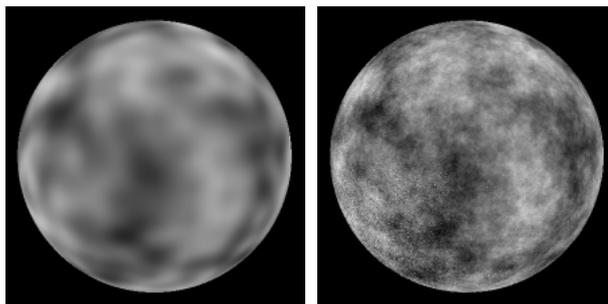
Frequency Gap. As frequency gap increases, the noise becomes more uniform. As frequency gap decreases, the noise becomes more lumpy.

Figure 14.62: fBm with Frequency Gap of 1.2 and 4



Octaves. Fewer octaves produce a less complex texture. More octaves produce a more detailed texture.

Figure 1 4.63: fBm with 2 and 16 Octaves



X Index. The local x coordinate of the point. For a discussion of local and global coordinates, see [Global Coordinates](#) on page 182.

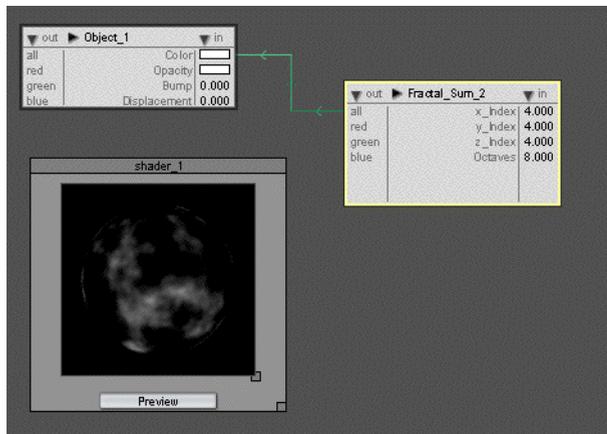
Y Index. The local y coordinate of the point.

Z Index. The local z coordinate of the point.

Fractal Sum

The fractal sum noise function. This is another type of well defined randomness.

Figure 1 4.64: Float Nodes>Fractal Sum



X Index. The local x coordinate of the point. For a discussion of local and global coordinates, see [Global Coordinates](#) on page 182.

Y Index. The local y coordinate of the point.

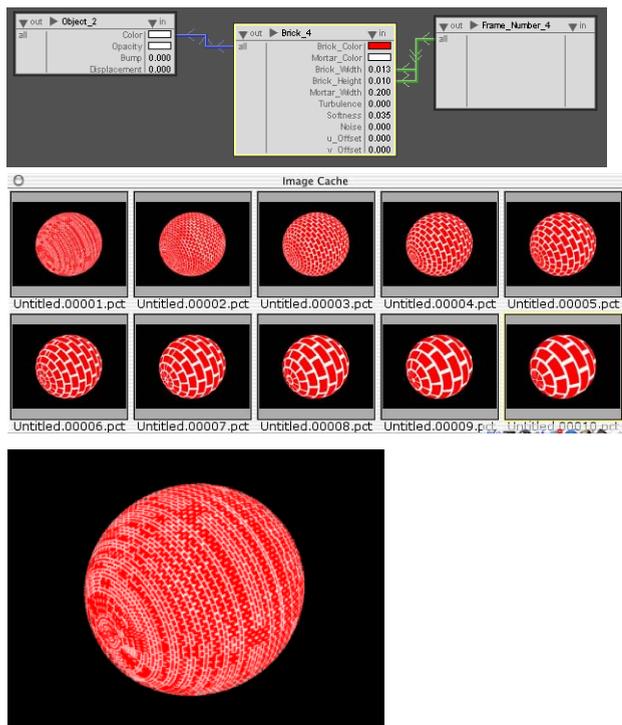
Z Index. The local z coordinate of the point.

Octaves. The difference between the octaves and the nearest powers of 2 determines the pattern. Larger powers of 2 are more complex (but after 8 the effect is subtle). Powers of 2 are: 2, 4, 8, 16, 32...

Frame Number

Returns the value of the frame number. For instance, in frame 0, this node returns 0. In 5 this node returns 5. In frame 90 this node returns 90. Use this to increase a parameter as an animation progresses. In the animation below, the **Frame Number** node is connected to the **Brick Height** and **Brick Width**.

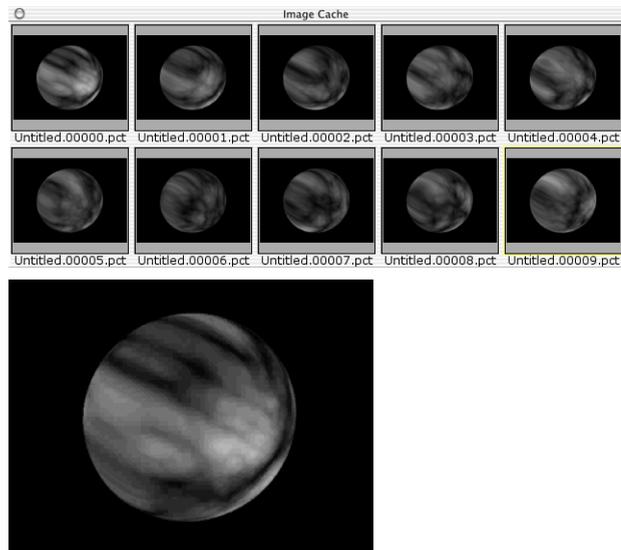
Figure 14.65: Float Nodes>Globals>Frame Number



(Click the picture to view the animation)

One very useful thing to do with **Frame Number** is attach it to *x*, *y*, or *z* in a noise function to get the pattern to move across your object. For instance, in the animation below, **Frame Number** is connected to *X* on a Turbulence node.

Figure 14.66: Frame Number Connected to X on Turbulence

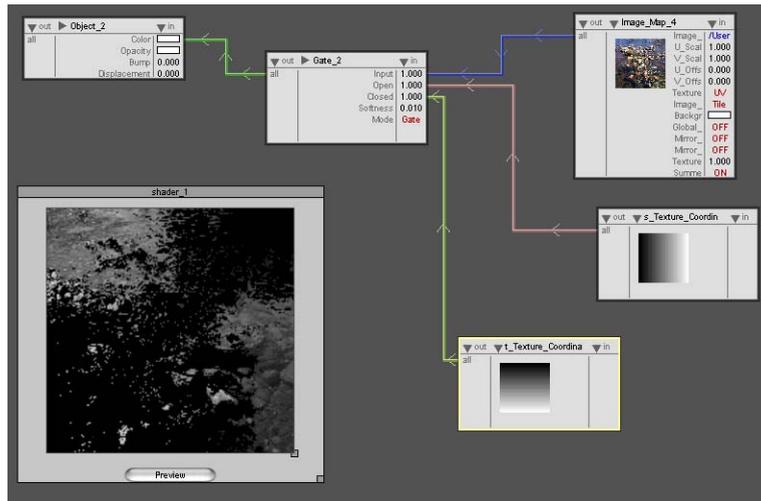


(Click the picture to view the animation)

Gate

This is similar to a luma key. It takes the input, and only allows values between the **Open** brightness and the **Close** brightness to pass through. In gate mode, it lets the specified brightness values through. In matte mode, it produces either black or white. Use small amounts of softness to control smoothing & falloff from the edges of the gate.

Figure 1 4.67: Float>Gate



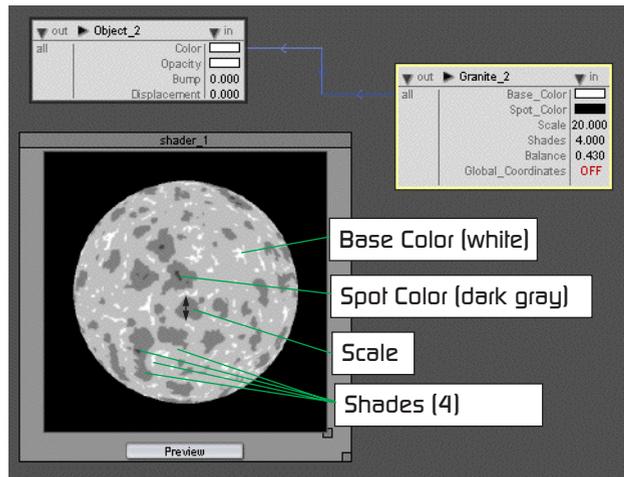
Input. Usually this is linked to an image map.

Open and Closed. These two values control which portions of the input pass through the gate.

Softness. Softens the edge between values that pass through the gate and value that do not pass through the gate.

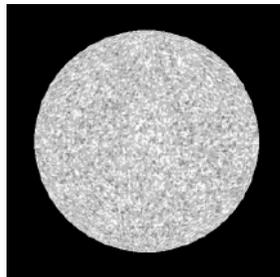
Granite

Figure 14.68: 3D Textures > Granite



Scale. The size of the spots.

Figure 14.69: Default Granite Scale (1)



Balance. The ratio of base color to spots.

Figure 14.70: Granite with Balance 0.25

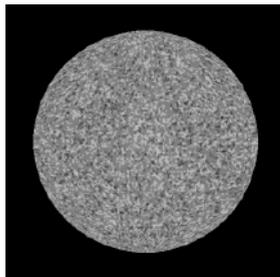


Image Map

Surfaces the object with an image map. See [Creating a Texture](#) on page 249 for a tutorial on creating an accurately positioned image map. See [Modeling Olympus Mons](#) on page 269 for a tutorial on using an image map as an elevation map to create terrain.

Figure 14.71: 2D Textures>Image Map

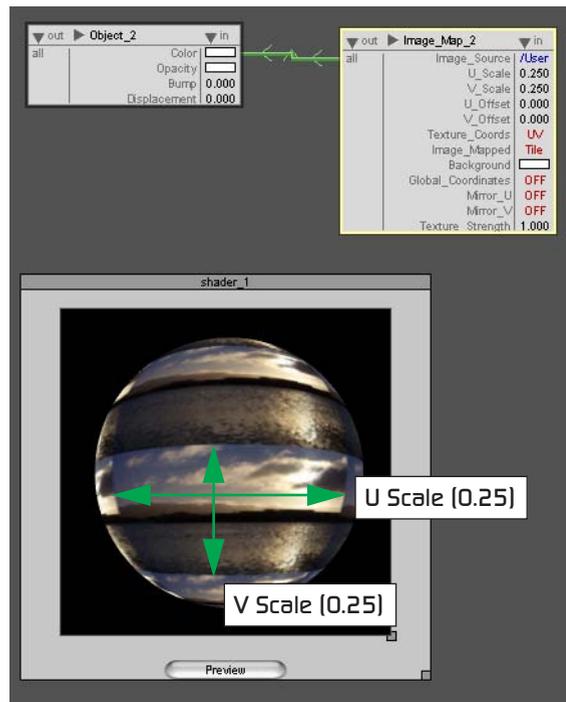


Image Source. Click on this field and select Load Texture Map from the resulting menu to load a new image map. If the image map you want to use is already loaded, select it from the menu. You can use any still image type supported by QuickTime. These include PICT, gif, jpg, tga, tiff, and png.

U Scale. Scale along the U dimension. A scale of 1 stretches all the way around a sphere. For a discussion of u and v see Periodic NURBS, U, and V on page 39.

V Scale. Scale along the V dimension. A scale of 1 stretches from one pole to the other on a sphere. For a discussion of u and v see Periodic NURBS, U, and V on page 39.

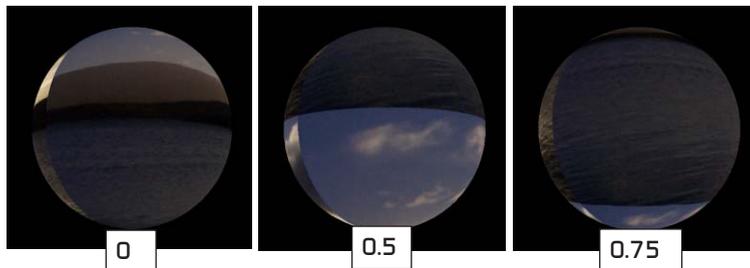
U Offset. The u distance between upper left corner of the image map and the beginning of the NURBS surface (the beginning of the NURBS surface is where u and v are 0). An offset of 1 pushes the image map completely around the NURBS surface so it is back where it started.

Figure I 4.72: Various U Offsets



V Offset. The v distance between upper left corner of the image map and the beginning of the NURBS surface (the beginning of the NURBS surface is where u and v are 0). An offset of 1 pushes the image map completely around the NURBS surface so it is back where it started.

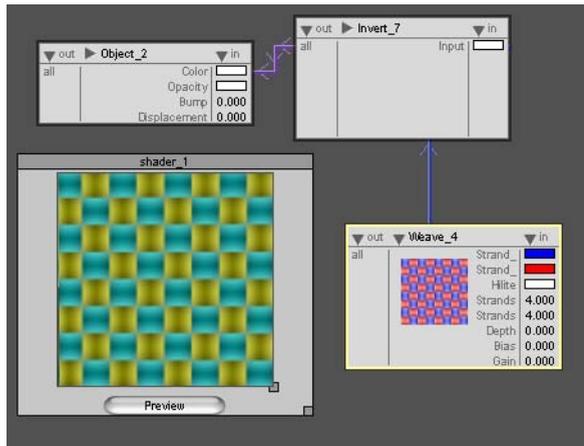
Figure I 4.73: Various V Offsets



Invert

Inverts the colors of the input.

Figure 14.74: Utility Functions>Invert

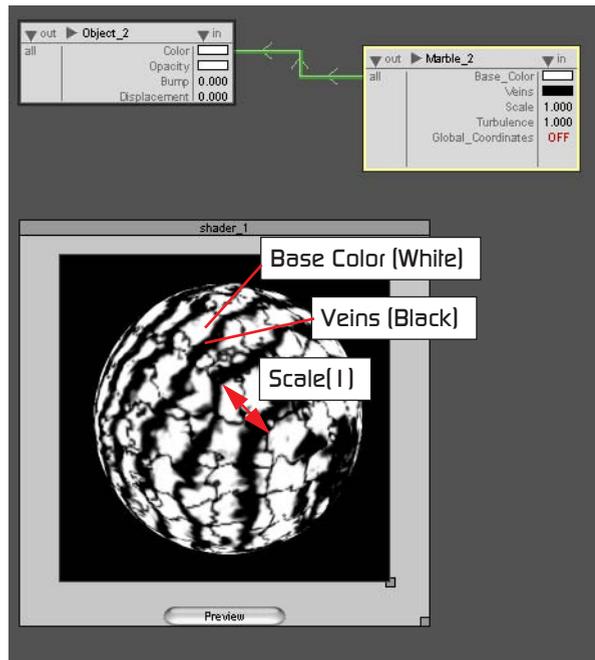


Input. Drag all from the out column of the source node to the Input of the Invert node.

Marble

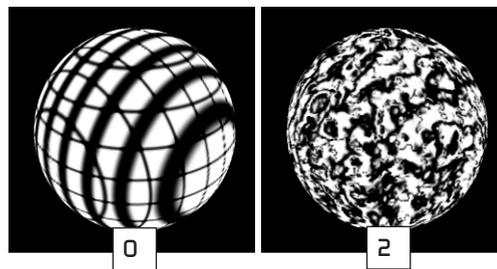
Intermixes two colors in a pattern much like that found in some natural marble.

Figure 1 4.75: 3D Textures>Marble



Turbulence. The amount of turbulence in the veins.

Figure 1 4.76: Marble Turbulence



Math Functions

The **Math Functions** node is one of the most powerful nodes in ShaderMaker Pro. There is so much that you can do with this one node that a whole book could be devoted entirely to math node techniques. This section illustrates a few of the things that you can do with the **Math Functions** node.

Figure 14.77: Float Nodes > Math [add]

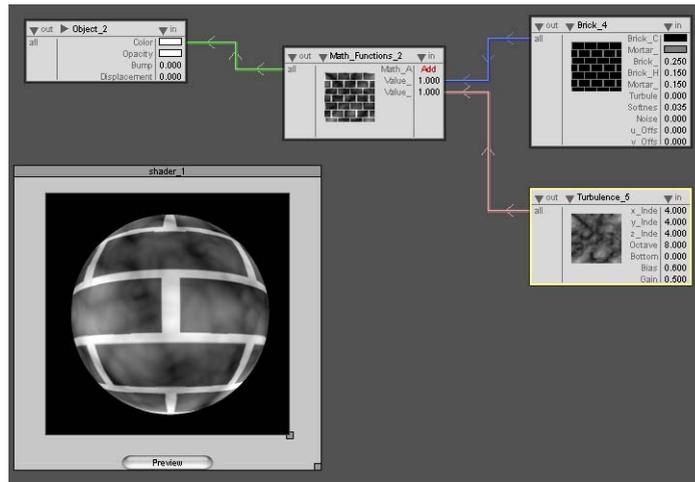


Figure 14.78: Float Nodes > Math [subtract]

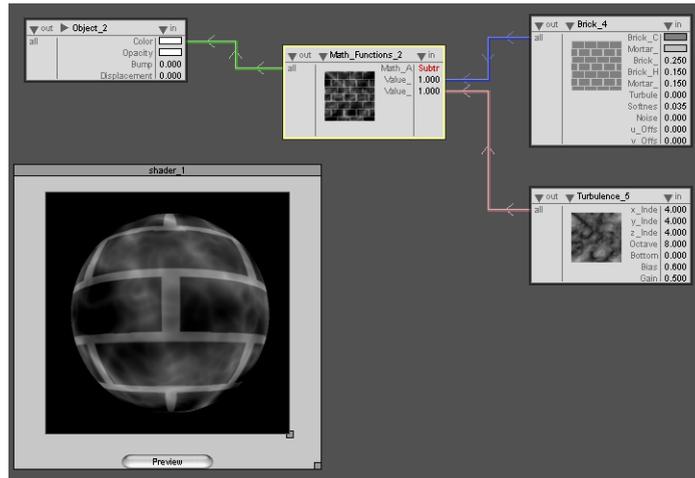


Figure | 4.79: Float Nodes>Math [*multiply*]

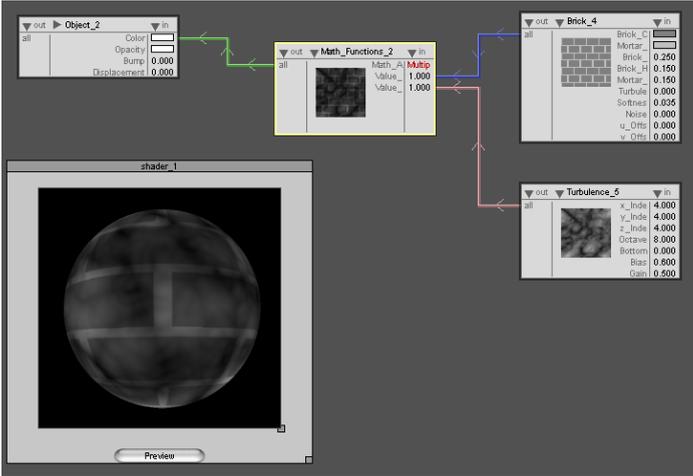


Figure | 4.80: Float Nodes>Math [*divide*]

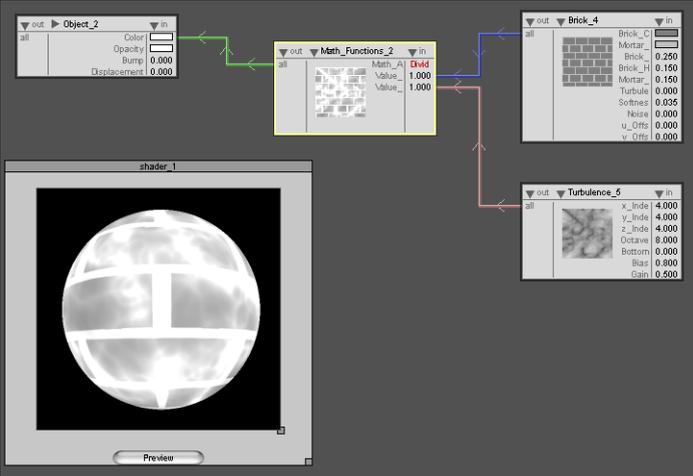
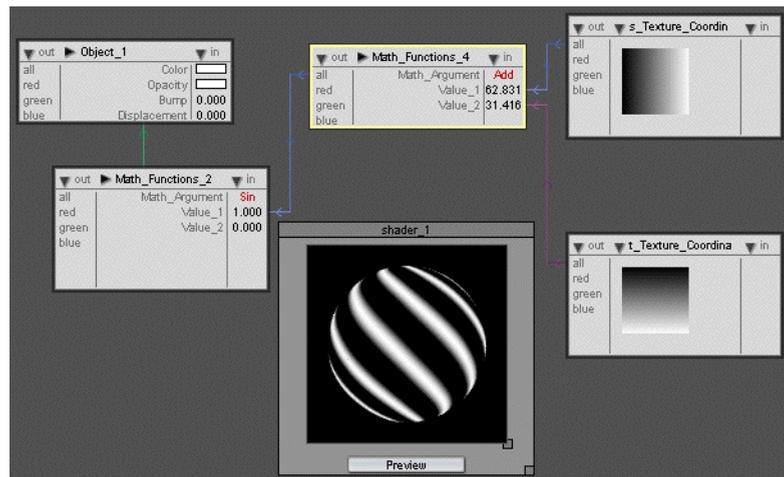


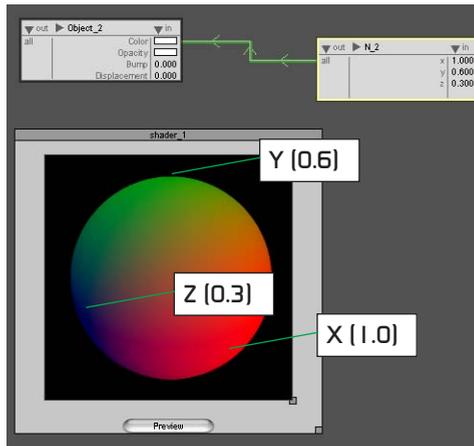
Figure 14.81: Float Nodes > Math [$\sin(62.8308*s+31.4159*t)$]

Note: The two constants above (62.8308 and 31.4159) are even multiples of π (3.14159). This is because the circumference of a sphere is $2*\pi*(\text{the radius})$. To get a sin wave to evenly fill the circle, the circumference must be an even multiple of one wavelength. Stated more mathematically, angles for trig functions are expressed in radians.

N

Normal. Colors the object according to the values of the x, y, and z normals. Surfaces perpendicular to x are colored red with an intensity equal to the x value. Surfaces perpendicular to y are colored green with an intensity equal to the y value. Surfaces perpendicular to z are colored blue with an intensity equal to the z value.

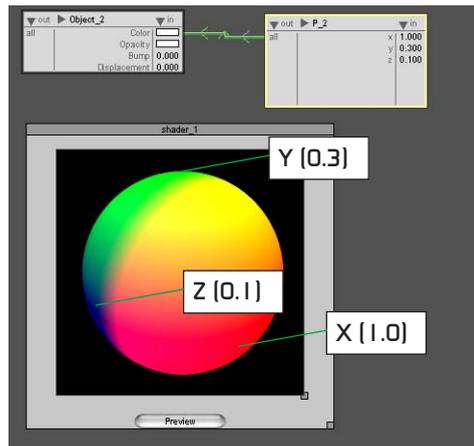
Figure I 4.82: Point>Globals>N



P

Surface point. Colors the object according to the distance of the surface point from the world origin. Positive X values are colored red, positive Y values are colored green, and positive z values are colored blue.

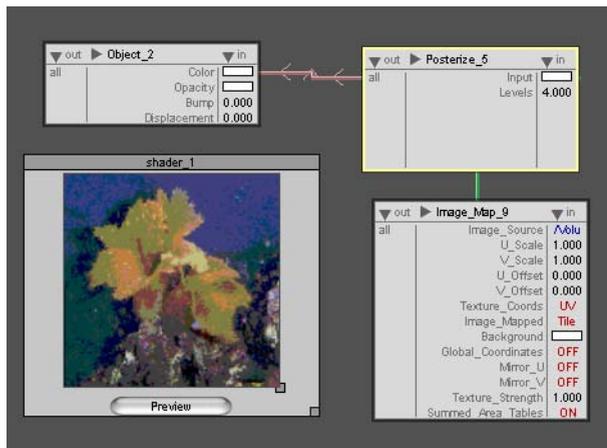
Figure I 4.83: Point>Globals>P



Posterize

Posterize takes an image and reduces the number of colors the image uses.

Figure 14.84: Utility Functions>Posterize



Here is the original image, which is specified under Image Source in the Image Map node:



(This is a *Sinularia dura* coral, often called a "flower leather" or "cabbage leather" coral.)

Input. The source image to be posterized. Drag all from the out column of the Image Map node to the Input of the Posterize node.

Levels. The number of levels of red, green, and blue in the posterized image. If your image is a full-color image, setting Levels to 4 gives you 48 colors maximum.

ProbeLight

ProbeLight is for illuminating your object with diffuse ambient light. The vectors together control things like where the light is coming from, how the lighting color is distributed, how bright, and how diffuse the light is. Individually, the vectors have no easily explainable physical definition.

The ProbeLight node is intended to be used with an unfinished application. You supply digital photographs of a chrome ball, and the application will give you vectors to simulate the lighting conditions in which that chrome ball was placed. More documentation will be available when the program which generates the vectors is finished.

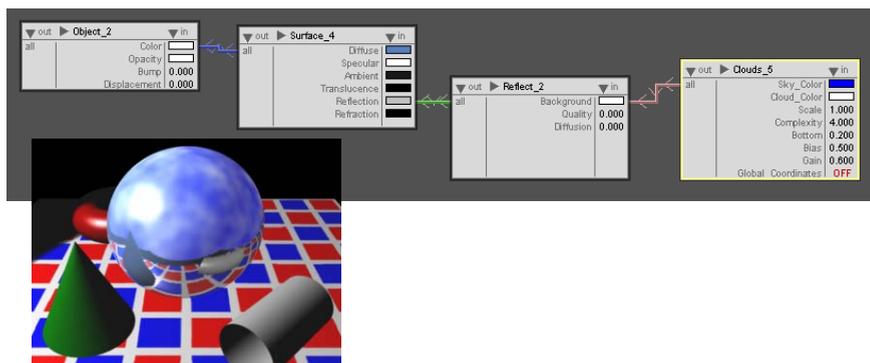
Ramp

This node is very similar to Color Ramp (See Color Ramp on page 202), except that it returns values, not colors.

Reflect

Plug the Reflect node into the Reflection input of a Surface or SM3Surface node.

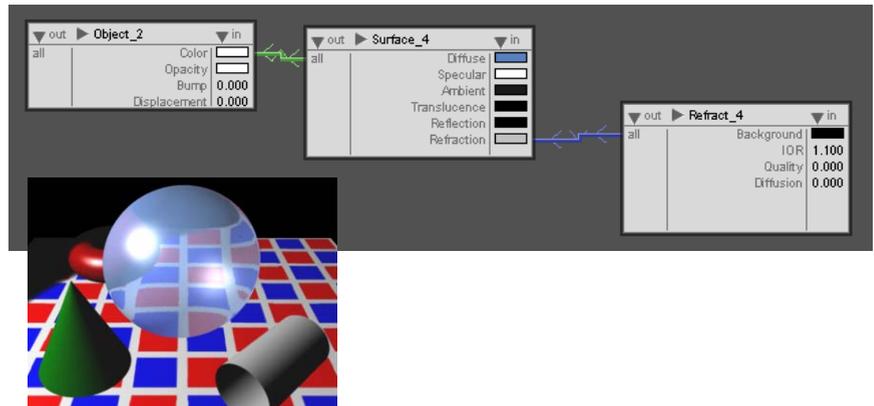
Figure 14.85: Raytraced Functions>Reflect



Refract

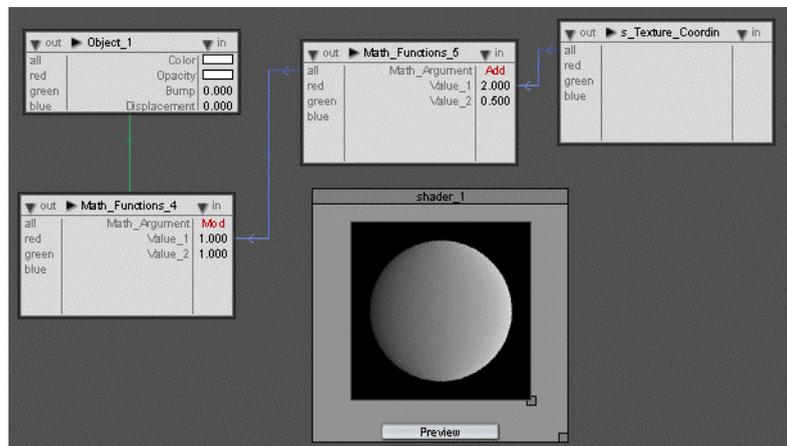
Plug the Refract node into the Refraction input of a Surface or SM3Surface node.

Figure 14.86: Raytraced Functions>Refract (IOR 1.1)

**S**

U surface coordinate. This ranges from 0 to 1 corresponding to the value of u. See Periodic NURBS, U, and V on page 39 for more information about u.

Figure 14.87: Float Nodes>Globals>S

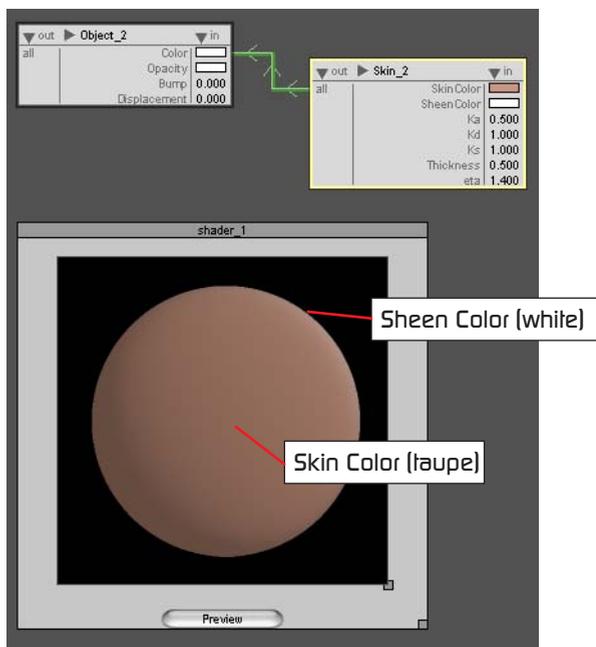


Note: The two math nodes are to rotate the coordinates on the preview so the picture better illustrates the meaning of s. S is the texture coordinate that corresponds to the u NURBS coordinate.

Skin

One of the basic surface nodes. Ka, Kd, and Ks are abbreviations for ambient, diffuse, and specular values (see Ka, Kd, and Ks on page 182). Thickness and eta are subsurface scatter values (see Subsurface Scatter on page 167).

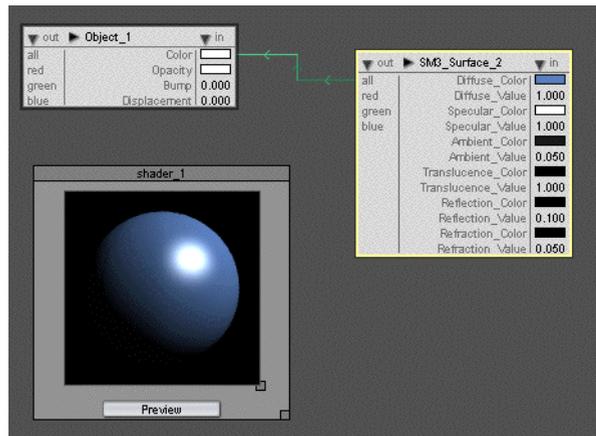
Figure 14.88: Basics > Skin



SM3 Surface

Shader Maker 3 Surface. This gives you the same surface inputs found in Shader Maker from PiXELS 3D 3.x. See Surface Nodes on page 178. For information about Reflection and Refraction see Raytracing on page 275.

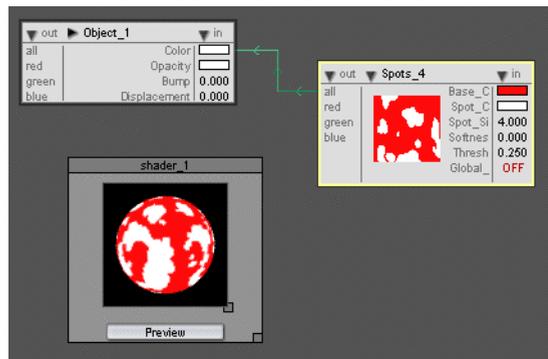
Figure 14.89: Basics>Shader Maker 3 Surface



Spots

Intermixes two colors in a blotchy or spotted pattern.

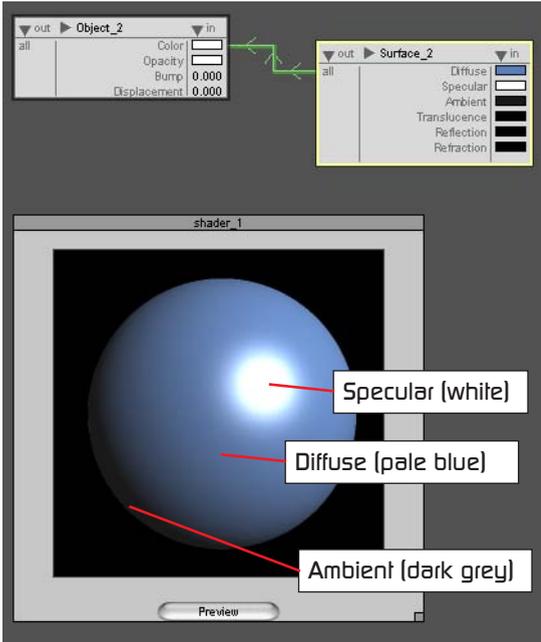
Figure 14.90: 3D Textures>Spots



Surface

A surface with Phong specular highlights. See [Surface Nodes](#) on page 178. For information about Reflection and Refraction see [Raytracing](#) on page 275.

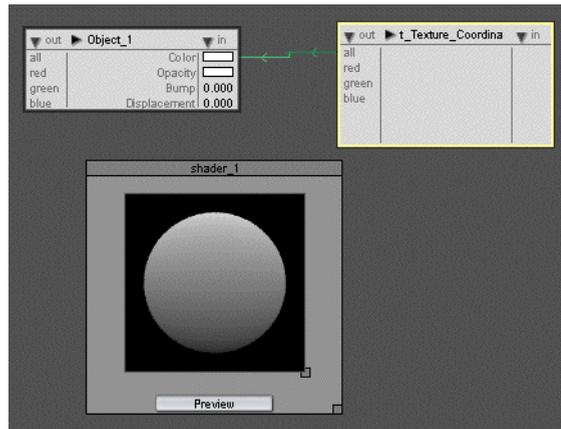
Figure 14.91: Basics>Surface



T

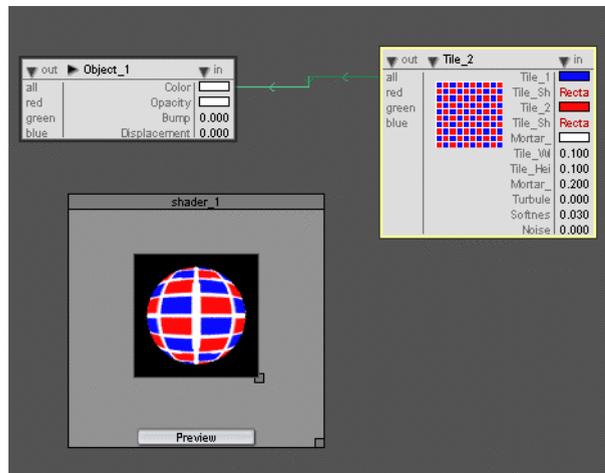
V surface coordinate. This ranges from 0 to 1 corresponding to the value of v. See Periodic NURBS, U, and V on page 39 for more information about v.

Figure 14.92: Float Nodes>Globals>T

**Tile**

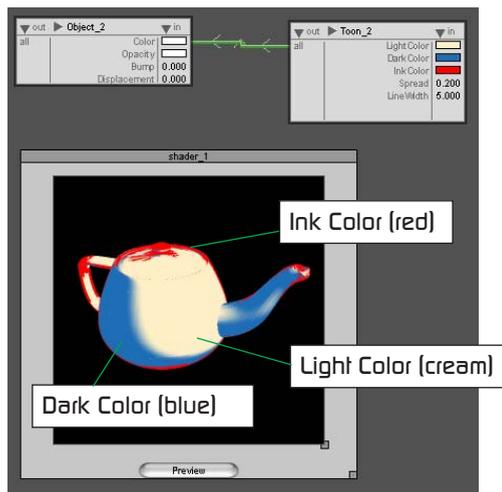
Tiles a surface with square or rounded tiles in alternating colors.

Figure 14.93: 2D Textures>Tile

**Toon**

Produces high contrast renderings with shadows somewhat like those found in cartoons.

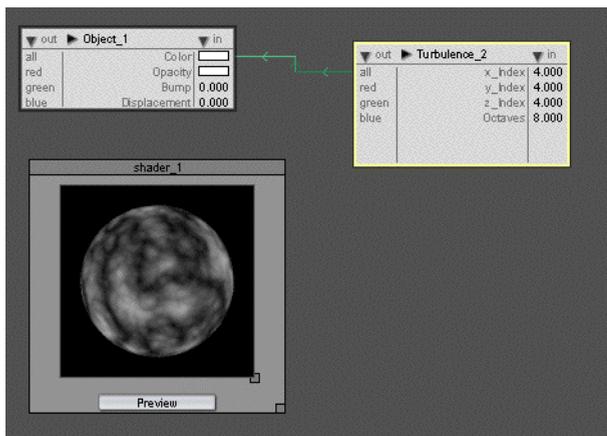
Figure I 4.94: Basics>Toon



Turbulence

A noise function that simulates the clumping of particles in a turbulent gas like a puffy cloud or smoke from a steam engine.

Figure I 4.95: Float Nodes>Turbulence



X Index. The local x coordinate of the point. For a discussion of local and global coordinates, see [Global Coordinates](#) on page 182.

Y Index. The local y coordinate of the point.

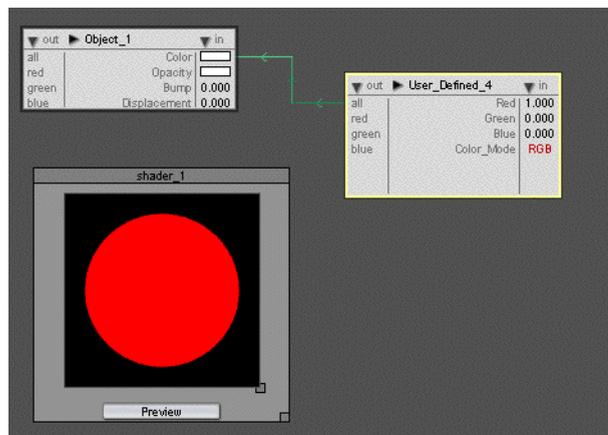
Z Index. The local z coordinate of the point.

Octaves. The difference between the octaves and the nearest powers of 2 determines the pattern. Larger powers of 2 are more complex (but after 8 the effect is subtle). Powers of 2 are: 2, 4, 8, 16, 32...

User Defined

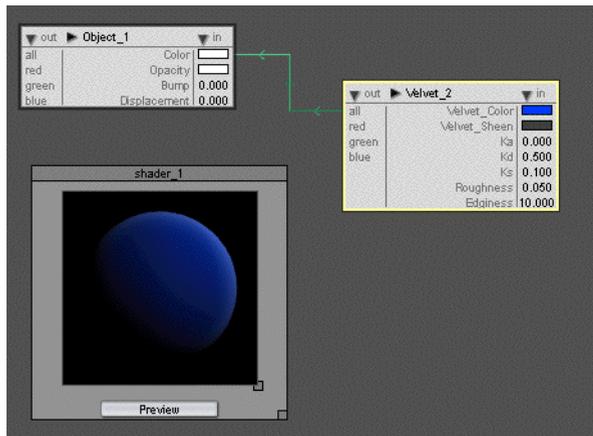
User defined color.

Figure 14.96: Color Nodes > User Defined



Velvet

Figure I 4.97: Basics>Velvet



Wave2D

Creates a 2D wave function.

Figure I 4.98: Float Nodes>Wave2D



U Pos. Sets the U position of the center of the wave function.

V Pos. Sets the V position of the center of the wave function.

To create stripes parallel to V, set U Pos to a high value and leave V Pos at 0.5. To create stripes parallel to U, set V Pos to a

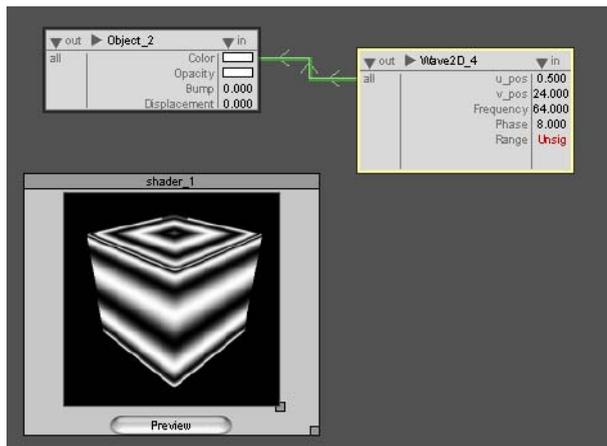
high value and leave **U Pos** at 0.5. If you set both **V Pos** and **U Pos** to high values, you will get diagonal stripes. The higher the value, the straighter the stripes will be.

Frequency. Sets the number of rings in the wave function.

Phase. Use this to control where the wave starts. Adjust this value to make the center of the wave white, black, or grey.

Range. Signed produces output in the range of -1 to 1. Unsigned produces output in the range of 0 to 1.

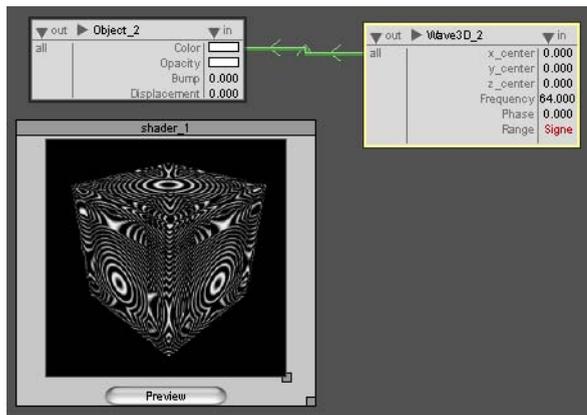
Figure 14.99: Wave2D Stripes



Wave3D

Creates a 3D wave function.

Figure 14.100: Float Nodes>Wave3D



X Center. Sets the x coordinate of the center of the wave function.

Y Center. Sets the y coordinate of the center of the wave function.

Z Center. Sets the z coordinate of the center of the wave function.

Frequency. Sets the number of rings in the wave function.

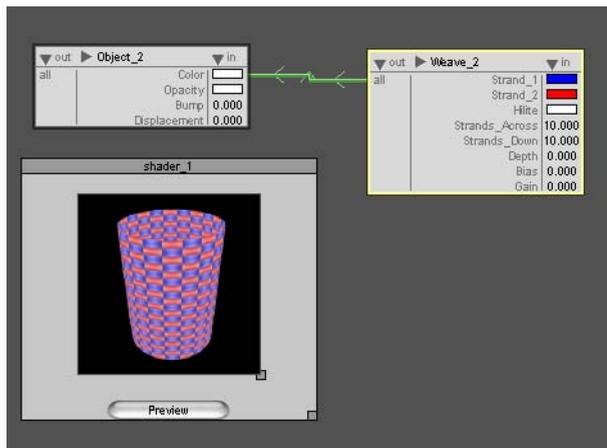
Phase. Use this to control where the wave starts. Adjust this value to make the center of the wave white, black, or grey.

Range. Signed produces output in the range of -1 to 1. Unsigned produces output in the range of 0 to -1.

Weave

Creates a 2D woven surface.

Figure 14.101: 2D Textures>Weave



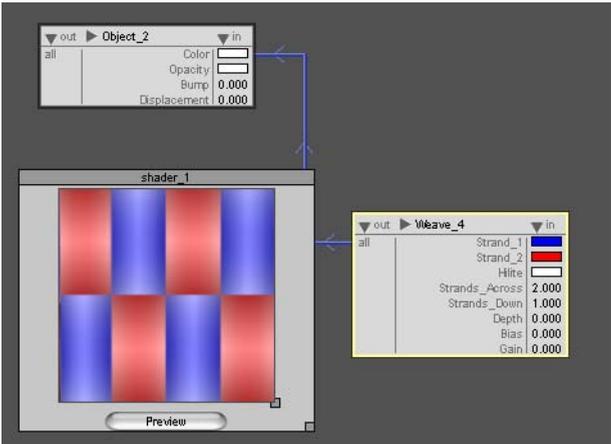
Strand 1. Color chooser for the U direction.

Hilitte. Multiplier for the strands: any non-black color tints the strands with this color.

Strands Across. Coarseness or fineness of the weave in the U direction. A value of 1 means one alternation across, resulting in two strands.

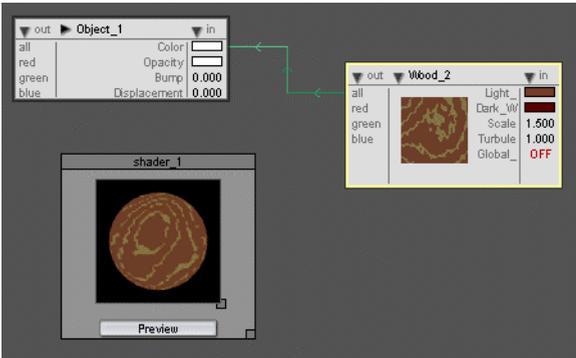
Strands Down. Relative coarseness or fineness of the weave in the V direction. A value of 1 means one alternation down, resulting in two strands.

Figure 1 4.1 02: Weave with 2 Strands Across, 1 Strand Down



Wood

Figure 1 4.1 03: 3D Textures>Wood



Chapter 15. Tutorial: Real Rock

Granite

In this tutorial we want to build a shader that looks like a piece of rock. Figure 15.1 shows a close-up photo of some granite. We want a shader that looks like this.

Figure 15.1: Granite

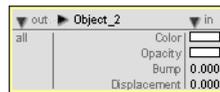


One way to do it would be to edit the edges of the photo so the image could tile across the surface without seams. A much cleaner solution, however, would be to create a procedural shader that describes the proper texture and lighting characteristics of the surface.

1. Click on the **Shading** layout icon () to work with shaders. In this layout ShaderManager and TextureManager are at the top, the Attribute Manager is on the right, ShaderMaker Pro is in the center, and the Timeline is at the bottom.

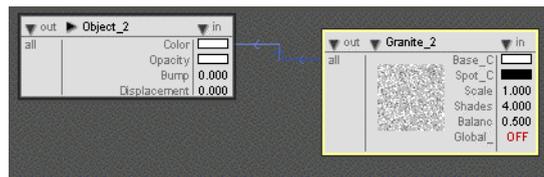
- Click on **New** in ShaderManager. This creates a new shader called `shader_1`. The new shader contains a single **Object** node.

Object Node



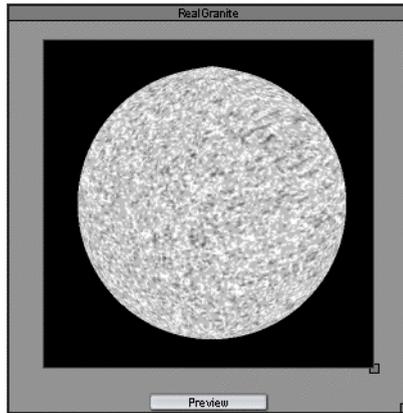
- Double-click on the shader name in ShaderManager, and change it to **RealGranite**.
- The speckled grey color is the most common, so focus on it first. Click on the **Nodes** button at the bottom of ShaderMaker Pro. This opens the node menu. Select **3d textures>granite** from this menu. This creates a **Granite** node.
- Click and drag the mouse from the area under **out** (marked **all**) on the **Granite** node to the white box next to **Color** on the **Object** node. This connects the **Granite** node to the **Object** node.

Granite Connected to Object



- Click **Preview** to see what the shader looks like. The texture is close, but the real granite you are working to recreate is horizontally stretched a little, is generally darker and more blue-grey, and has irregular brown or rust colored areas. Also, the current shader has no shadowing or highlighting.

Preview of Granite

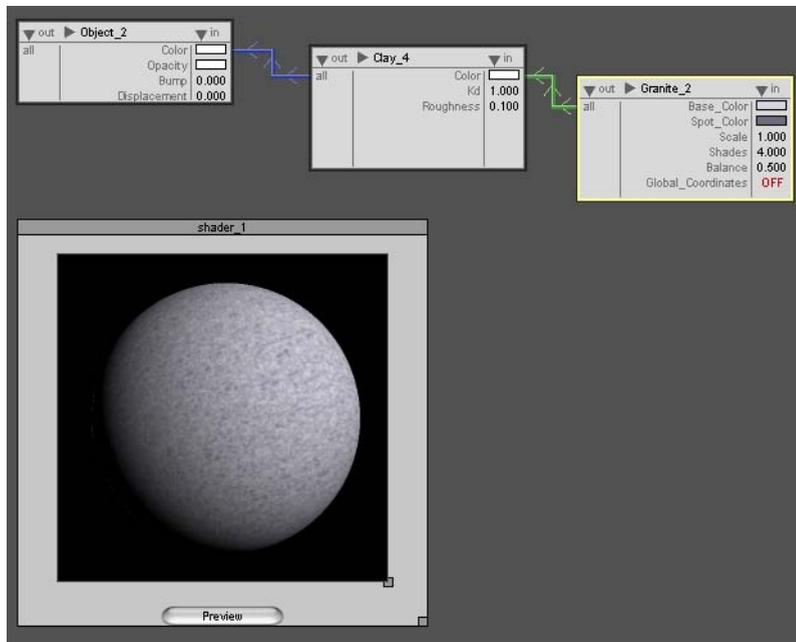


To add shadowing and highlighting, you need to add surface nodes. Surface nodes include: **Surface**, **SM3_Surface**, **Skin**, **Velvet**, **Clay**, and **Diffuse**. Since the next level roughness on the granite is very clay-like, use a **Clay** node.

If the surface was shiny or plastic-like, you would use **Surface** or **SM3_Surface**. If the surface had a waxy or skin-like appearance, you would use **Skin**. If the surface had a velvety appearance, you would use **Velvet**.

1. From the node menu select **diffuse functions > clay**. This adds a **Clay** node.
2. Drag from the output of the **Clay** node to the **Color** input of the **Object** node.
3. Drag from the output of the **Granite** node to the **Color** input of the **Clay** node.
4. Click on the **Base Color** sample (the white rectangle in the column next to **Base Color**) in the **Granite**. This opens a color selector. Choose a light purplish-grey (in the color selector choose **HSV** then choose the following values: 262 Hue Angle, 5% Saturation, 88% Value).

- Open the color selector for the **Spot Color** on the **Granite** node and choose a darker blue-grey (239 Hue Angle, 16% Saturation, 51% Value).



To add the color accents, you need to add noise and blend the noise with the main pattern. First, add the rust color.

- Add an **fBm** node: from the **Nodes** menu choose **float nodes>fbm**.
Note: **fBm** stands for "fractal Brownian motion", since it is derived from the mathematics used to describe Brownian motion, which is the random motion of atoms.
- To stretch the pattern into elongated streamers, set the **x Index** field to **30** and set the **z Index** field to **16**.
- To give the pattern more contrast, set the **Gain** to **0.8**.

4. To see what the resulting noise looks like, click the triangle next to the name of the **fBm** node.

Since you want the light and dark parts of the pattern to produce different colors, you can use a **Blender** node.

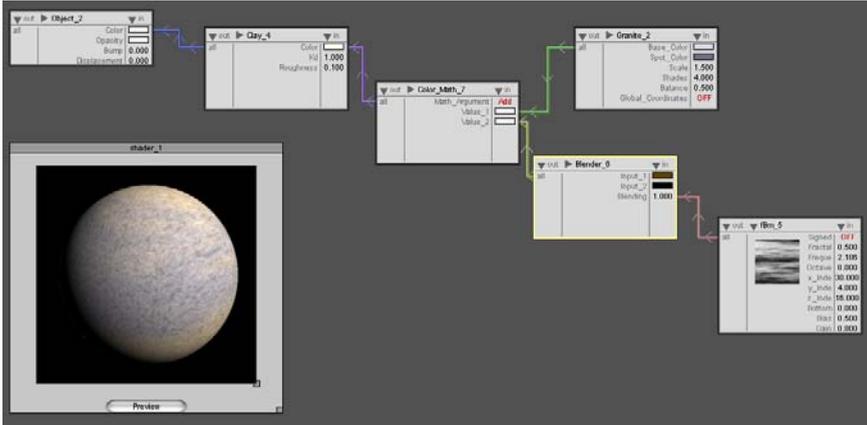
1. Add a **Blender** node: from the **Nodes** menu choose **utility functions>blender**.
2. Set **Input 1** of the **Blender** node to a brown color (Hue Angle 41 , Saturation 99%, Value 37%).
3. Plug the output of the **fBm** node into the **Blending** input of the **Blender** node.
4. Set the **Blending** value to 1.
5. To see the resulting rust color pattern, click the triangle next to the name of the **Blender** node.

Next, you want to mix the rust color pattern with the speckled pattern of the rock. The **Color Math** node lets you combine patterns with the **Add** function.

1. Add a **Color Math** node: from the **Nodes** menu choose **color nodes>color math**.
2. Make sure the **Math Argument** field is set to the **Add** function.
3. Plug the output of the **Granite** node into **Value 1** of the **Color Math** node.
4. Plug the output of the **Blender** node into **Value 2** of the **Color Math** node.
5. Plug the output of the **Color Math** node into the **Color** input of the **Clay** node.

Now click the **Preview** button to see the results.

Figure 1 5.2: Realistic Granite



Chapter 16. File Types

There are a wide variety of file types you need to be aware of when dealing with any 3D software.

Scene Files

A PiXELS 3D 4 scene file contains all the information in your scene except for the shader specifications. Shaders are saved in as separate files in a subdirectory called **Shaders**. Textures that are used in the shader are saved in a subdirectory called **Textures**. The following options from the **File** menu handle PiXELS 3D 4 scene files.

New. Starts a new file that only contains a single camera and a light.

Open. Opens a previously saved scene file and all the associated shader files.

Save. Saves your scene in PiXELS 3D 4 (.piv) format. If you are saving a new file, PiXELS creates a folder which contains subfolders that contain the different parts of your project. This includes a folder called **Shaders** which contains each shader in your project saved as a separate file.

Save As. Opens a file dialog so you can specify a new name or location for the scene file. As with Save, each shader is saved as a separate file in the **Shaders** folder.

Polygonal Model Files

PiXELS can import and export polygonal model files as 3D Studio (.3ds) files, Autodesk (.dxf) files, Lightwave Object (.lwo) files, and Wavefront (.obj) files.

To transfer a model from another 3D application...

1. Export the file from that 3D application to one of the supported formats. For DXF and OBJ, if you are given the choice between exporting as a binary or ASCII file, choose ASCII. If you are given a choice between exporting as polygons or curves, choose polygons.
2. Import the file into your scene. The imported object is a polygonal model that can be transformed or edited like any other polygonal object.

The following options from the **File** menu handle polygonal import and export.

Import>*filetype*. Places the polygonal geometry from the file in your scene. The reader only reads polygonal models. Materials and curved surfaces are ignored.

Export>*filetype*. Prompts you to select a polygonal object. Exports the object as a polygonal model in the proper format.

Render Files

RIB is an industry standard format for specifying scenes to production quality renderers. This format was initially invented by Pixar Animation Studios to log sequences of Renderman calls. The format is now used to describe scenes to Renderman compliant renderers. (Tempest, the renderer that comes with PiXELS 3D is nearly Renderman compliant.) When you select **Render** from the **File** menu, PiXELS 3D saves a RIB file in the `resources>ribs` folder under the PiXELS 3D 4.0 folder. PiXELS then launches Tempest which queues and renders each RIB file.

Instead of selecting **Render** and rendering your scene immediately, you can export the scene as a RIB file, then launch Tempest and load the file manually. This allows you to render

many files as a batch process or allocate more memory to Tempest (and render more ambitious scenes).

The following options from the **File** menu handle render files.

Import>RIB Scene. Imports lights, cameras, and geometry from the RIB file. If you want to move a scene from PiXELS 3D 3.7 to 4.x, export the scene as RIB in 3.7, then import the Scene in 4.

Import>RIB Object Archive. Imports geometry only. If you want to move an object from one PiXELS 3D 4 scene file to another, export it as RIB then import it as a RIB Object Archive.

Export>RIB. Exports the current scene in RIB format. Like piv format, RIB format contains everything in the scene except for shader specifications. Shaders are saved as separate files.

Render. Exports the current scene in RIB format and renders it in Tempest.

Shader Files

Shaders tell the renderer how an object interacts with light at any given point on the object's surface. Each shader in a scene is saved as a separate file when the scene is saved, when the scene is rendered, or when the scene is exported as a RIB file. Shader files are always saved in ShaderMaker Pro (.smp) format.

Most of the shader file handling options are accessed from ShaderMaker or ShaderManager in the Shader Layout.

ShaderManager

New. Creates a new shader with a single **Object** node. You can then create other nodes and attach them to the **Object** inputs to construct your shader.

Load. Opens a file dialog so you can load an existing shader into your scene.

Delete. Removes the selected shader from your scene.

ShaderMaker

File>Save. Saves the current shader.

File>Save As. Opens a file dialog so you can specify a new location or name for the current shader.

Texture Files

Textures are bitmap files that are used in shaders. You can load Photoshop (.psd), BMP, JPEG (.jpg), PICT (.pct), PNG, TIFF (.tif), QuickTime images, or Targa (.tga) files as textures.

Most of the texture file handling options are accessed from **ShaderMaker** or **TextureManager** in the **Shader Layout**.

TextureManager

Delete. Removes the currently selected texture from the scene.

Load. Opens a file dialog so you can load an existing shader into your scene.

ShaderMaker Image Map Node

If you click on the **Image Source** input in the **Image Map** node in **ShaderMaker**, the **Image Source** menu opens. The following file handling options are on this menu.

Load Texture Map. Opens a file dialog so you can load an existing shader into your scene.

Delete Texture Map. Removes the texture from the scene currently selected for the node from the scene.

Tag Maps

In vertex mode you can select vertices on a NURBS object and export a tag map to help you figure out where to draw different things on your texture map. See **Creating a Texture** on

page 249 for information on using a tag map to create a texture.

Export>Tag Map. Exports a Tag Map of the selected NURBS object. A tag map is a regularly spaced rectangular grid corresponding to the uv grid on the NURBS object. Selected vertices are depicted as intersections with dots. Unselected vertices are depicted as intersections without dots.

Rendered Files

You can render your scene to a wide variety of bitmap formats. In addition, you can render an animation to a sequence of bitmaps, then assemble the bitmaps into a movie with the Flipbook program included in the `Tools/PIXELS/Mikros Img` subfolder in the application folder.

Still Images

You can set the format, size and other attributes of the rendered file in the Attribute Manager under **Render Options>Image Setup**. There, the following options are available.

Name. The rendered file gets saved with the specified name followed by a `.` , followed by the frame number followed by a `.` and a 3 character extension determined by the file format. The file gets saved in the `resources/ribs` sub-folder in the application folder.

For example, if you name the file `Arthur`, and render the initial frame in PICT format, the file gets saved as `Arthur.0.pct`.

Height. The height of the bitmap in pixels.

Width. The width of the bitmap in pixels.

Pixel Aspect Ratio. The squareness of the pixels. An aspect ratio of 1 produces square pixels.

File Format. Tempest can render to the following formats: SGI image, Photoshop (.psd), BMP, JPEG (.jpg), PICT (.pct), PNG, TIFF (.tif), QuickTime images, or Targa (.tga).

File Content. The information contained in each pixel in the rendered file.

rgb. Red, green, blue file. In this kind of file, each pixel contains values for the three colors.

rgba. Red, green, blue, alpha file. In this kind of file, each pixel contains values for the three colors plus a value for the transparency, or alpha of the pixel. This is the typical file content for final renderings.

z. A depth or z buffer file. Each pixel contains information about the object distance from the camera. If the camera is placed at the same location as a light, the z-buffer image can be used as input to the shadow buffer for that light on a subsequent render.

rgbz. Red, green, blue, z file. Each pixel contains values for the three colors plus a value for the depth or z of the pixel.

rgbaz. Red, green, blue, alpha, z file. Each pixel contains values for the three colors plus a value for the transparency, plus a value for the depth of the pixel.

Animations

To generate a sequence of images for an animation, in the Attribute manager select **Render Options > Animation > Render Animation**, and then choose a **Render** command from the **File** menu. This will create a series of images with the same name as the **pix scene** file, followed by a "." and the frame number, with the extension appropriate to the file type specified in **Render Options > Image Setup > File Format**. These will be created in the **renders** folder in your project's file folder (For example,

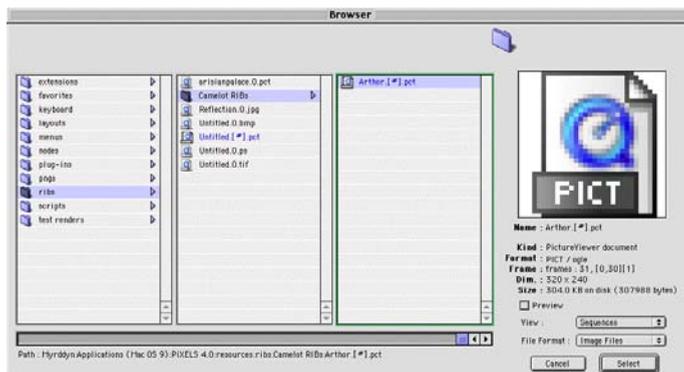
if your PiXELS3D project file is named `myscene.piv`, the images will be in the `myscene/renders` folder.)

To assemble the individual rendered frames into a movie, either use the `Flipbook` program that comes with PiXELS 3D or use Apple's `QuickTime Pro`.

The `Flipbook` program is located in the `tools/PIXELS/Mikros Img` sub-folder in the PiXELS3D application folder. Launch the program and choose `Open a Sequence` from the `File` menu. This opens a file browser for choosing an image sequence. Go to the `renders` folder in your scene folder and select the image sequence. An image sequence is shown in blue as a single entry in the file browser, even though the sequence is composed of many files.

For example, if there were 30 PICT files rendered to the file name `Arthur`, the files would be called `Arthur.0.pct` through `Arthur.29.pct`. These files would show up in the `Flipbook` browser as `Arthur.[#].pct`.

Figure 16.1: Flipbook File Browser



When you select the sequence and click `OK` you get an options dialog where you can specify various characteristics of the

resulting animation. After adjusting any options you want, click **OK**.

Note: If only part of your animation loads into **Flipbook**, you have not allocated enough memory to the program. Exit **Flipbook**. Then, from the **Finder**, **Get Info** on the application file and set the **Preferred Memory** as high as available RAM will allow. Relaunch **Flipbook** and your image sequence should load.

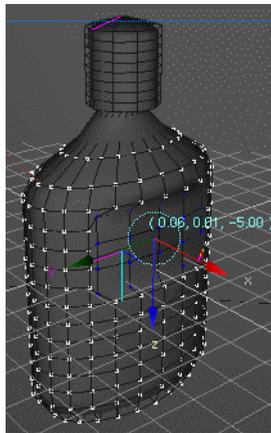
Once the animation is in **Flipbook** you can play it in different playback modes and watch the frames per second (**fps**) achievable on your computer. When you are through testing the movie, select **Export to Movie** from the **File** menu.

Adjust the compression and quality settings as appropriate and click **OK** to save your animation. The movie is saved in QuickTime (**.mov**) format.

Chapter 17. Tutorial: Creating a Texture

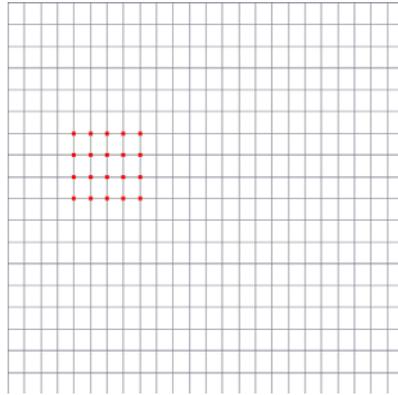
Suppose you want to draw a label for a bottle, and you want the image of the label to get correctly positioned on the bottle. To do this, you will first export a tag map, then you will use the tag map to help you draw your image.

1. Select the vertices on the bottle that coincide with the location of the label.



2. From the menu select **File>Export>Tag Map**.
3. Save the tag map file and then open it with a 2D graphics program like Adobe Photoshop, Macromedia Fireworks, or Procreate Painter. The tag map shows all the uv grid

lines on the NURBS object with the selected vertices indicated as red points.



4. If your graphics program supports layers, add a layer and draw the label over the area indicated by the red dots.
5. Hide the layer with the grid and export the image. Use a format like .png that supports alpha channel transparency.

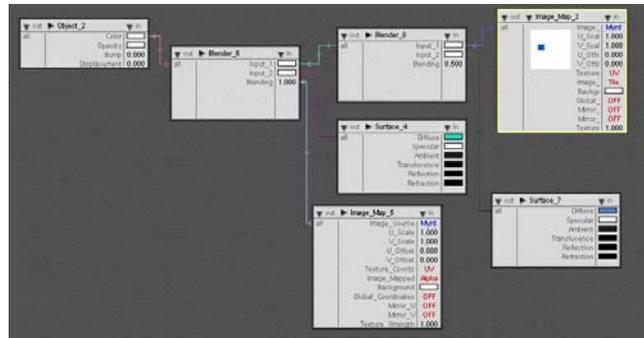


6. In PiXELS, select the bottle and create a new shader by clicking **New** in **Shader Manager**.
7. Create an **Image Map** node by clicking on **Nodes** in Shader-Maker and selecting **2D textures > image map** from the resulting menu.

8. Connect the Image Map node to Color by dragging from out on the Image Map to Color on the Object node.
9. Click on the field under in next to Image Source on the Image Map node and select Load Texture Map from the resulting menu.
10. Choose the label image you created and render. The label should be in the right place.



11. To give your bottle highlighting and shadowing, you can use Blender nodes to mix in Surface nodes.



With shaders added to the background and the cap, you can render your final image.

Figure 17.1: Finished Bottle

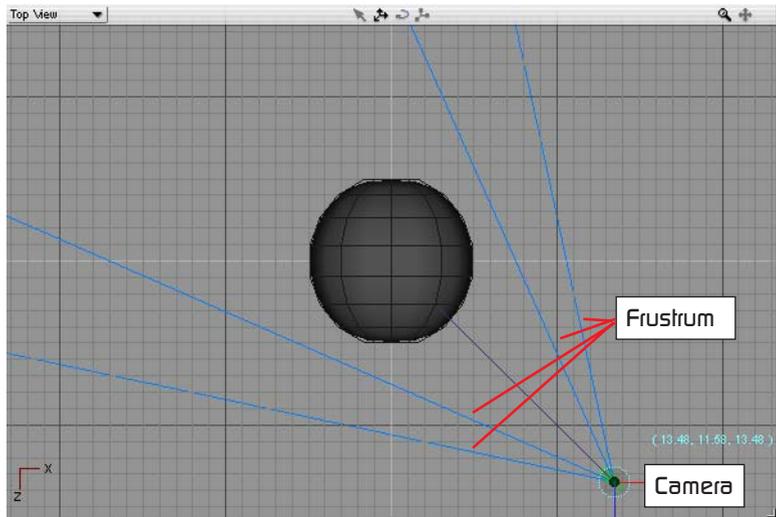


Chapter 18. Cameras and Composition

Camera Parameters

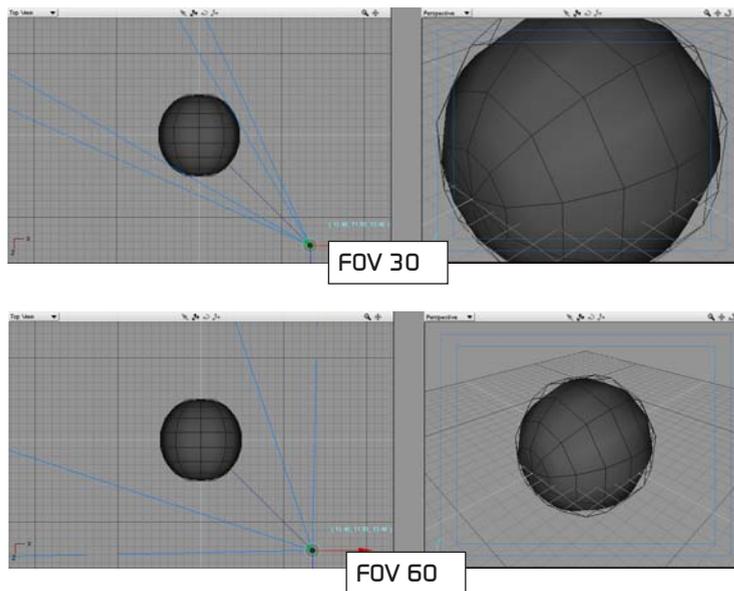
Show Frustrum. The *frustrum* is the outline of what the camera can see. When you show the frustrum a graphic is drawn in the views to show what can be seen by the camera.

Figure 18.1: Frustrum Shown



FOV. Field of view.

Figure 18.2: Two FOVs



Using Multiple Cameras

To add a camera to the scene,

Choose **Special > Camera** from the main menu.

To switch the view to a different camera...

1. Choose **Select Camera** from the view menu (the menu in the corner of each view).
2. When prompted select the camera you want to use.

To render with a different camera,

In the Attribute Manager open the **Render Options** attribute group and choose the **Rendering Camera** from the drop-down list.

Director camera

You do not have to render from every camera you use. A camera that you do not render from is called a director camera. A director camera allows you to look around your scene from different angles, much as a director would walk around a the-

atrical set. A director camera is used in modeling and animating the scene, but is not used for the final rendering (although there is nothing preventing you from changing your mind and rendering from a camera you had been using as a director camera). You can use multiple director cameras without affecting your final rendering view. You can move the director cameras as desired, whether it is to get a close-up view of an intricate model, to follow one object's animation, or to get a feel for the overall composition of a scene. You can also use a director camera to test view camera positions in an animation.

View camera

The View camera determines the final view of your scene that will be rendered. You can move the view camera during animations. For example, you could create a walk through of a scene.

Composition

The goal of a composition is to direct the viewer's eye to the subject and keep the viewer interested in the image. Whether the subject matter is beautiful or gruesome, no one will look at it for long unless the composition creates a pleasing, interesting, or striking image.

This chapter is not a course in composition; rather, it includes a few brief refresher paragraphs and some comments that are particularly applicable to 3D scenes and animations.

The Rule of Thirds

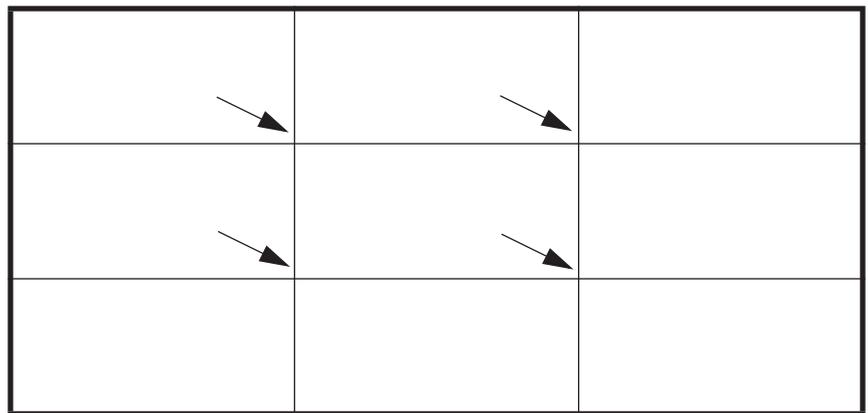
The Medieval Era: Center the Subject. Medieval paintings look static and posed partly because of the way the artists composed their paintings. Medieval rules of composition had the subject at the center of the painting, with supporting, secondary subjects arrayed around the primary subject.

The Renaissance and Beyond: the Golden Section. During the Renaissance, the rules of composition changed somewhat. The golden section is more frequently found in paintings. The

golden ratio is derived mathematically, but is approximately equal to 1:1.618. Da Vinci, a mathematician as well as an artist (and inventor), used it often. He was probably the first person to call it the golden section, though it was used by the Greeks in constructing the Parthenon. Vermeer's *The Painter's Workshop* uses a web of golden section. Mondrian also used it extensively.

Current: Rule of Thirds. The Rule of Thirds is based on the golden section, but it is much easier to use. You can approximate the golden section by comparing a rectangle to one that is two-thirds the height and two-thirds its width. This leads to the generalized rule of thirds: place a subject along one of the lines that divide the rectangle into thirds. Ideally, the focal point of the image would be along both a horizontal and a vertical division line. In the illustration below, focal points are indicated by arrows.

Figure 18.3: Focal Points Determined by the Rule of Thirds



Drawing Attention to the Subject

Life. If an object in a scene is alive, it will automatically become the subject if there is no other life. If several things in the composition are alive, and there is no other life, the group as a whole becomes the subject.

Movement. If an object moves, attention will be drawn to that object. This includes both actual movement, as in an animation, and implied movement, which is used in still images.

Contrast. One of the best ways to indicate the subject is with contrast. An excellent example is the proverbial bull in the china shop. The bull contrasts with the china shop in several different ways.

Uniqueness. The bull draws attention to itself because it is the only bull in among hundreds of pieces of china.

Color. While the china is uniformly white, the bull is a dark color.

Several years ago, there was a cartoon that featured an animal detective and his human partner. Naturally, the human played the sidekick. The animators employed an interesting technique to focus the viewers' attention: all of the animal characters were painted in bright, saturated colors, while all of the human characters were painted in subdued, greyed colors. Similarly, numerous TV commercials have used a colored subject among a background of black-and-white objects.

Size and Visual Weight. The large bull completely overpowers the pieces of small, lightweight china. Even the shelves creating the china displays have a light, airy feeling.

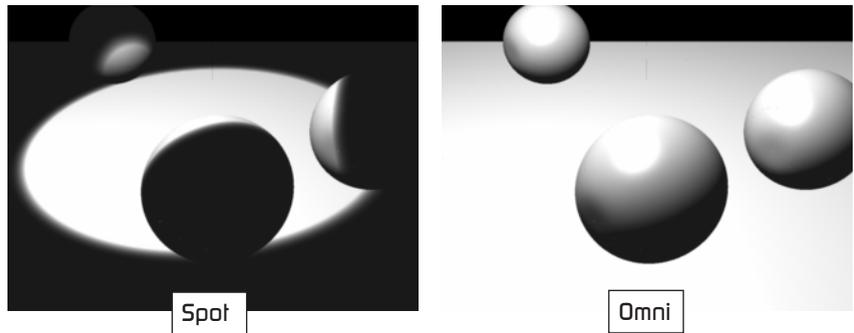
Shape and Form. While the china is composed mainly of curves, and the shelving is composed of straight lines, both china and shelving are regular shapes: you could construct the lines that make the shapes with a compass and straight edge. The bull, however, is an irregular form, composed of lines that are neither straight nor smooth curves.

Chapter 19. Lights and Lighting

Light Parameters

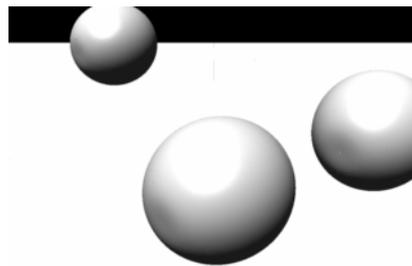
Light Type. A Spot light provides a cone of illumination. An Omni light casts light in all directions.

Figure 19.1: Light Types



Parallelrays. Illuminates the scene as if from an infinitely distant light source.

Figure 19.2: Parallelrays On

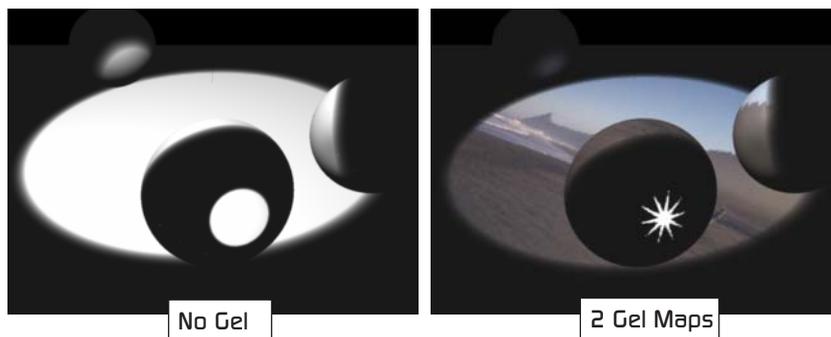


Intensity. The brightness of the light.

Color. The color of the light. Values should range from 0 to 1.

Gel Map. To project an image with a light, specify the image as a gel map. A gel map can be used like a slide projector to project an image or pattern, or it can be used like a mask to control the shape of a spot light.

Figure 19.3: Gel Maps



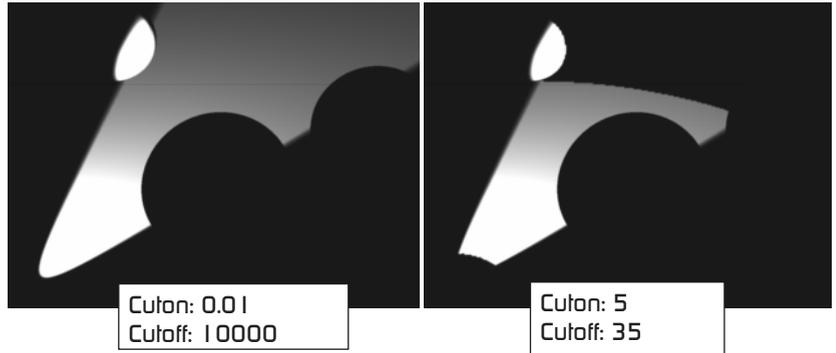
Always Show Cone. The light cone shows what will be illuminated if the light is a spot light. When Always Shows Cone is on, the cone is always shown in the views. When this is off, the cone is only shown when the light is selected.

Falloff.

Cuton. Distance from the source where the light starts.

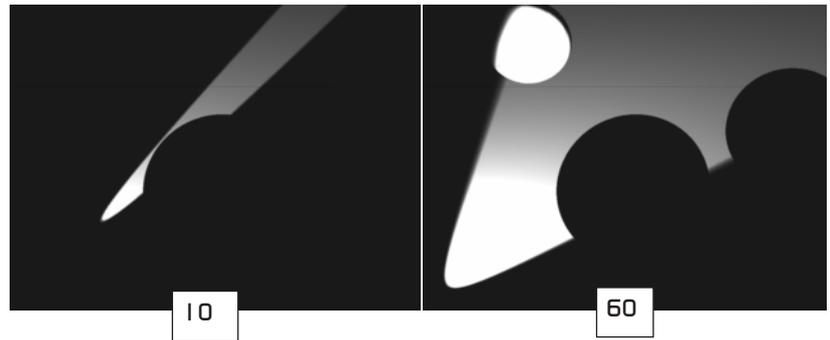
Cutoff. Distance from the source where the light ends.

Cuton and Cutoff

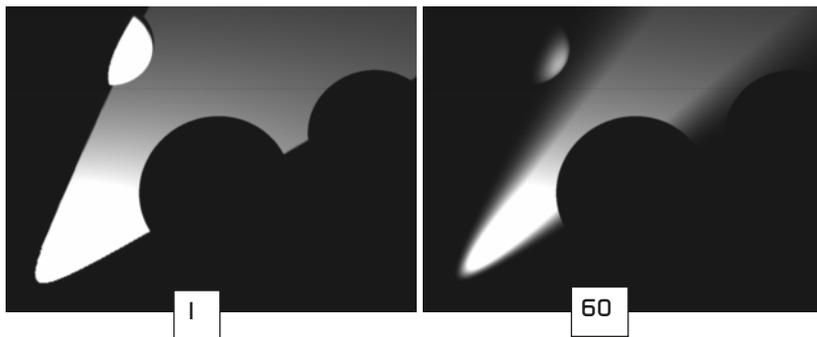


Spot Light Options.

Cone Angle. The width of the light cone. This can range from 1 to 180 .

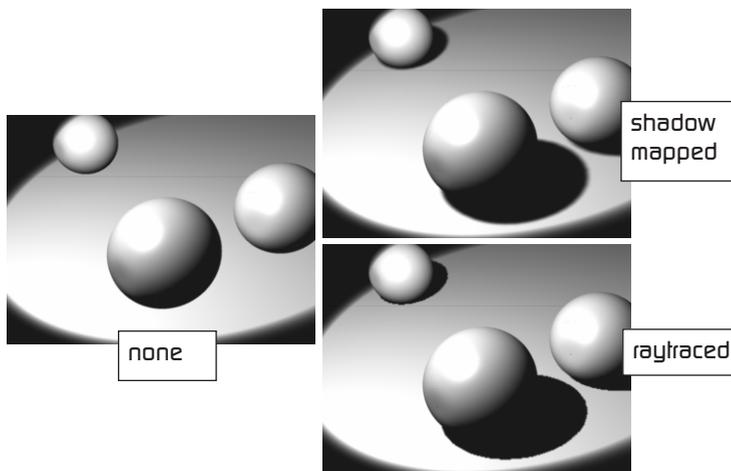


Penumbra Blur. The fuzziness of the light cone.



Shadows. Individuals objects with surfaces have shadows regardless of the Shadow Type chosen for the lights. However, to cast shadows from one object across another, you need to select Raytraced or Shadow Mapped shadows for the light.

Shadow Type. The way shadows are calculated.



Softness. Makes the edges of shadows blurry. Soft shadows on a shadow mapped shadow have fixed amount of blur all the way around the edge of the shadow. On ray-

traced shadows the amount of blur depends upon both the blur setting and the distance the shadow falls from the object which cast the shadow. In other words the softness on raytraced shadows is calculated as if the light source is an area light.

Chapter 20. Rendering

To render your scene,

Select **Quick Render** or **Final Render** from the **File** menu. This saves a RIB file and all the shaders from your scene in the `/ribs` sub-folder of your project folder. Then, the file is queued for rendering in Tempest, which opens and renders the file. For details about the files, see **Render Files** on page 242. The rendered image is saved in the `renders` subfolder of your project folder.

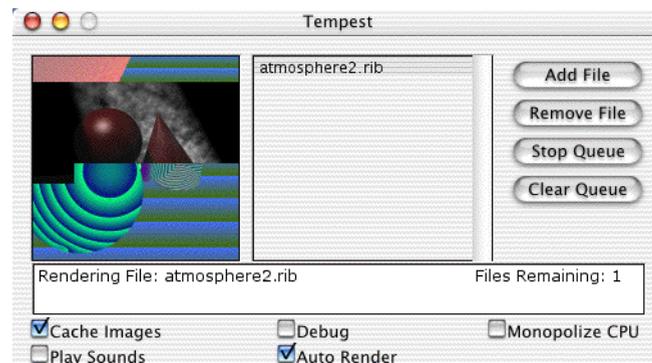
Quick Render. Cmd-r. Renders your scene with render parameters set to optimize render speed. Quick renders are always 320 by 240 and have no anti-aliasing.

Final Render. Cmd-f. Renders your scene with the settings specified under **Render Options** in the **Attribute Manager**.

Tempest

The main Tempest window shows the current state of the image being rendered, the files on the queue, the current status, and the render controls.

Figure 20.1: Main Tempest Window



Cache images. Stores rendered images in a viewable cache. Up to 10 images are viewable. Thumbnails of the images are shown in the Image Cache window. Click on a thumbnail to view it full size.

Figure 20.2: Image Cache Window



Play Sounds. Plays a sound clip when the rendering completes or encounters an error.

Debug. Shows more verbose status messages.

Auto Render. Starts rendering anything in the queue automatically. If you have Auto Render on, and for some reason you do not want the queued files to start rendering when you open Tempest, delete the file `Queue.dat` from the PiXELS 3D application folder before opening Tempest.

Monopz CPU. Monopolizes the CPU so your image gets rendered faster, but you cannot do other things on your computer while the image renders.

Sky Dome

Adds subtle colored lighting to the scene to make objects appear as if they are illuminated from the sky.

Light Map. The bitmap that you supply to Sky Dome affects the color of the lighting. So, for instance, if you supply a solid blue sky dome, the objects will appear as if they are under a clear, blue sky. If you supply an orange and blue sky dome, the objects will appear as if they are under a sunrise or sunset sky.

Exposure. The brightness of the sky.

Quality. The higher the quality, the more realistic your scene will look, and the more time it will take to render. So, insanely high quality will look really, really good but will take an insane amount of time to render.

Shadows. Enables/disables the casting of shadows. Render times are significantly longer when Shadows is enabled.

Max Distance. The maximum distance an occluder (shadow casting object) can be from the point being shaded and still contribute to the total shadow. Objects further away than Max Distance will have no effect on the shadowing of the point being rendered.

General Render Options

Pixel Samples. Width (x) and height (y) of pixels sampled to determine anti-aliasing. To disable anti-aliasing, enter 1 for these values. The default value is 3.

Shading Rate. The approximate number of micro-polygons per pixel. Usually a shading rate of one is sufficient. However, with intricate patterns, thin lines, or extreme displacement you may need a higher shading rate to avoid rendering artifacts. Higher shading rates take longer to render.

Bucket Size. A rendering is processed in square chunks called buckets. A larger bucket size will render faster, but require more memory.

Grid Size. Maximum number of micro-polygons allowed per grid unit.

Two Sided. Renders both sides of surfaces. Without this selected only one side of surfaces get rendered. This is faster, and often works well, however, with partially transparent shaders or objects like open cylinders where you can see both sides of a surface, you want to render two sided. If parts of objects are missing or if things on a surface appear backwards (as if you are seeing the inside of the back), check to make sure Two Sided is checked.

Dice Binary. Lines up micropolygon boundaries to avoid cracks.

Eye Splits. Controls how the renderer determines what parts of an object are in or out of the camera view. If you get the message "Cannot split primitive at eye plane" or if square chunks of an object near the camera are missing, increase the Eye Splits. Larger Eye Splits allow objects to get closer to the camera, but take a much longer time to render.

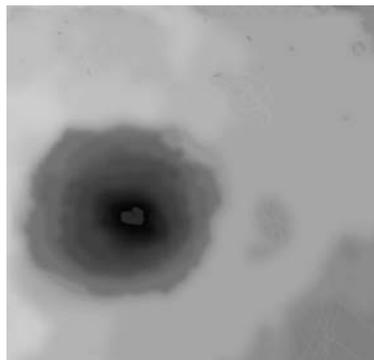
Chapter 2 I. Tutorial: Modeling Olympus Mons

Olympus Mons is an ancient volcano in the Tharsis region on Mars. It is one of the largest known mountains in the solar system, rising about 24 Km above the surrounding plain and stretching over a diameter of about 600 Km. (For comparison, Mauna Loa, the large volcano in the Hawaiian islands, rises about 14 Km above the ocean floor and about 9 Km above sea level.)

Recently the Mars Orbiter Laser Altimeter (MOLA) aboard the Mars Global Surveyor spacecraft took detailed height measurements of the entire planet. You can find the results of those measurements at <http://ltpwww.gsfc.nasa.gov/tharsis/mola.html>.

To do a model of a mountain, you need some height data. I've done a black and white drawing that approximates the height data for Olympus Mons. In this artistic interpretation, black represents the highest points and white represents the lowest points.

Figure 21.1: Height Map



You can use any height data you want, or you can use third-party software to generate a height map.

Terrain From Shaders

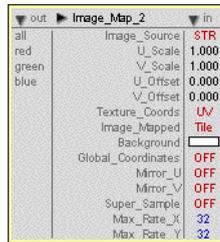
1. From the menus select **NURBS>Mesh** to create a mesh.
2. Click on the  (shading layout) icon or select **Layout>Shading** from the menus so you can work in ShaderMaker.
3. In the **ShaderManager** click **New**. This creates a new shader called `shader_1` with a single node called the **Object** node.



Image Map Node

1. Click **Nodes** or **Control-Click** in **ShaderMaker** to open the **Nodes** menu.

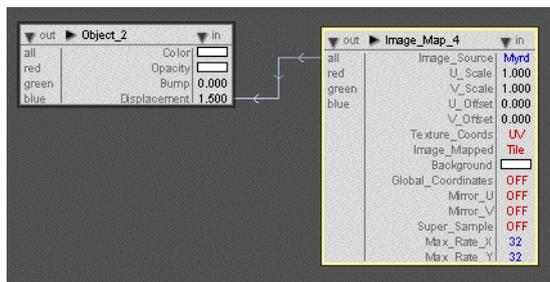
- From the **Nodes** menu select **2D textures>image map** to create an **Image Map** node.



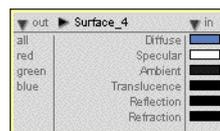
- Click on the **rst** item (**Image Source**) under **in** on the **Image Map** node. This opens the **Image Source** menu.
- Select **Load Image** from that menu.
- In the resulting **Open** dialog, find your height map and open it. (The height map of Olympus Mons is called **olympus.jpg** and can be found in the **PiXELS3D:Samples:Textures** folder.)

Displacement

- Click and drag from the area under **Out** on the **Image Map** node to the **Displacement** input on the **Object** node.
- Increase the **Displacement** value to 1.5.



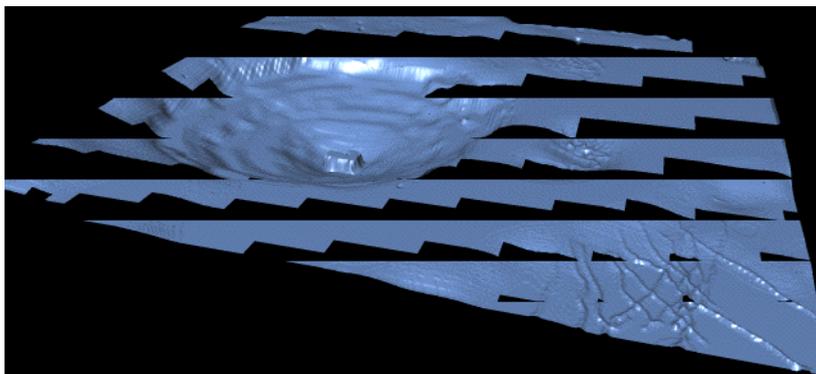
- From the **Nodes** menu select **basics>surface** to create a **Surface** node.



4. Drag from the area under out on the **Surface** node to the **Color** input of the **Object** node.

Now save your work and do a test render of your scene.

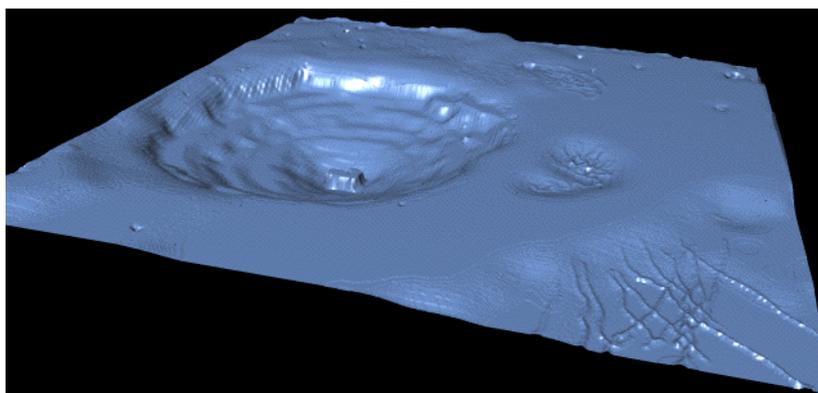
Figure 21.2: Out of Bounds Displacement



This does not look like a mountain. The problem is, the surface got displaced beyond the renderer's expectations.

Displacement Bounds

1. Go to the Attribute Manager. Under the **Mesh** object under **Objects** there is a list of **Render Options**. One of the **Render Options** is **Displacement Bounds**. Change the **Displacement Bounds** to 1.5 and render again.

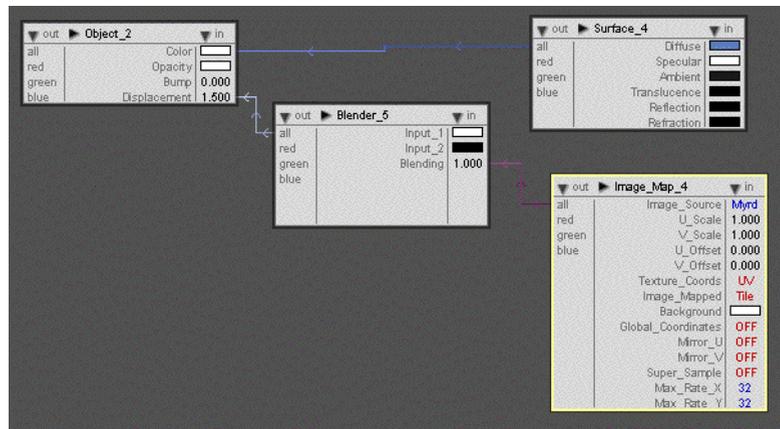


Now the black gaps are gone, but it looks like a big pit, not a mountain. In a displacement shader, the highest points are white and the lowest points are black.

You could open the image in an image editor and make a negative of your height map, or you could fix the problem in ShaderMaker. This tutorial fixes the problem in ShaderMaker.

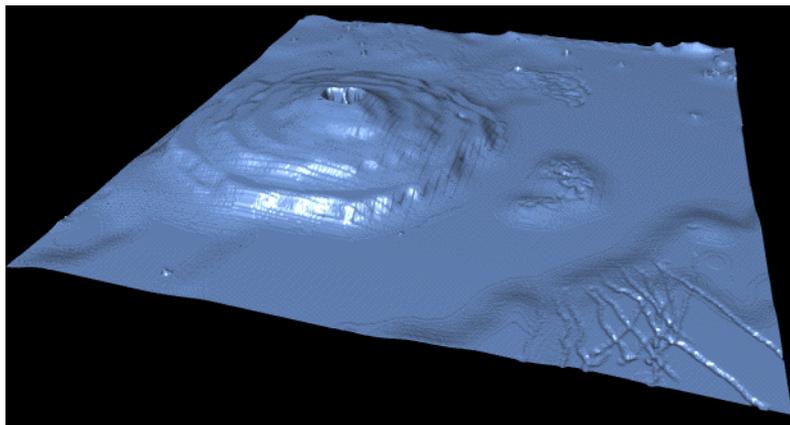
Making a Negative with a Blender

1. From the **Nodes** menu select **utility functions>blender** to create a **Blender** node.
2. Plug the **Image Map** node into **Blending** on the **Blender** node.
3. Plug the **Blender** node into **Displacement**.
4. Set the **Blending** value to 1.



Now save again and render.

Figure 2 1 .3: Olympus Mons with Simple Surface



Chapter 22. Raytracing

Raytracing is a technique of simulating light by calculating how individual light rays will bounce off or refract through an object. Raytracing is great for producing realistic reflective or transparent surfaces. Raytraced shadows are great for producing the kind of crisp shadows you get from focused spotlights.

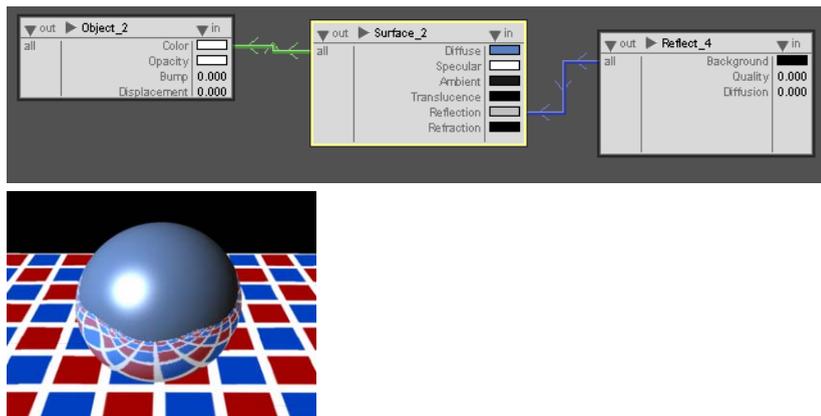
Reflection

To get a raytraced reflective surface...

1. Use a Surface or SM3Surface node along with a Reflect node.
2. Plug the Reflect node into the Reflection input of the Surface or SM3Surface.
3. Click on the Reflection input color and set it to something other than white or set the Reflection value on the SM3Surface to something less than 1. (If the Reflection color is not changed from white, the reflected image on the object will overwhelm the surface characteristics of the object.)

Reflection

Simple Reflection

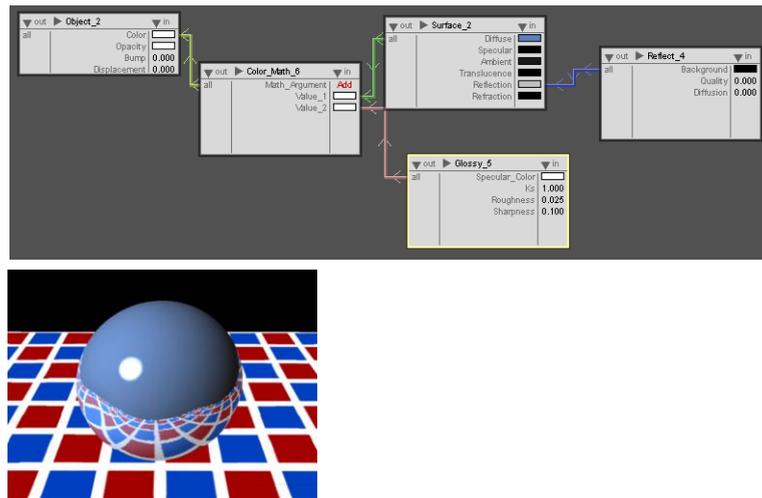


Specular Highlights

The default Phong specular highlights on a Surface or SM3Surface is an approximation of the way a diffuse surface reflects the light source. Raytraced reflections simulate shiny, rather than diffuse surfaces. When you have a raytraced reflection you should disable the Specular input on the Surface or SM3Surface by making it black or giving it a value of 0. You should then either use a hard-edged specular function node like Glossy or Specular, or you should make a simple model of the light source for your object to reflect.

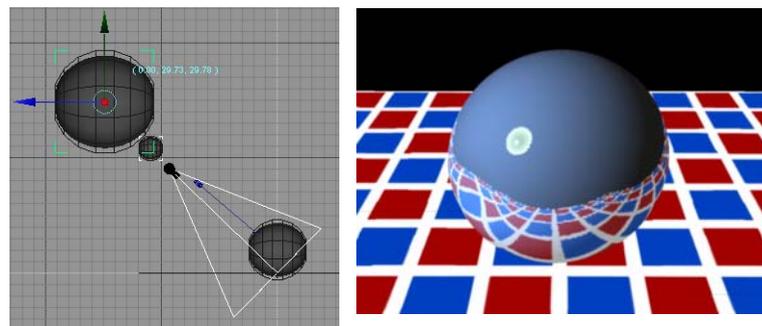
Glossy Node. To use the Glossy (or Specular) node connect the output of the Surface (or SM3Surface) to Input 1 of a Color Math node. Connect the Glossy node to Input 2 of the Color Math node. Set the function to Add. This adds the specular component to the other Surface components.

Figure 22.1: Glossy Highlight



Modeling a Light Source. When you model a light source just to create its reflection, remember, the only place this model is going to be seen is in the reflections on reflective surfaces. You can keep the model very simple and still have a very realistic scene. You can make the object(s) in the light model glow by not adding any nodes to their shaders. You just set the color on the Object node to the color you want.

Figure 22.2: Modeled Highlight



Background

When you are doing a scene without reflective surfaces, you only need to model whatever is directly in the camera view. When you have reflective surfaces, they will often reflect areas behind the camera or otherwise outside the part of the scene you modeled.

One way to fill in these otherwise empty parts of the reflection, is to plug a node into the Background input of the Reflect node. Depending upon your needs, you can use a procedural node, or you can draw or photograph a bitmap and plug the image into the reflection background through a Image Map texture node.

Figure 22.3: Procedural Background (clouds)

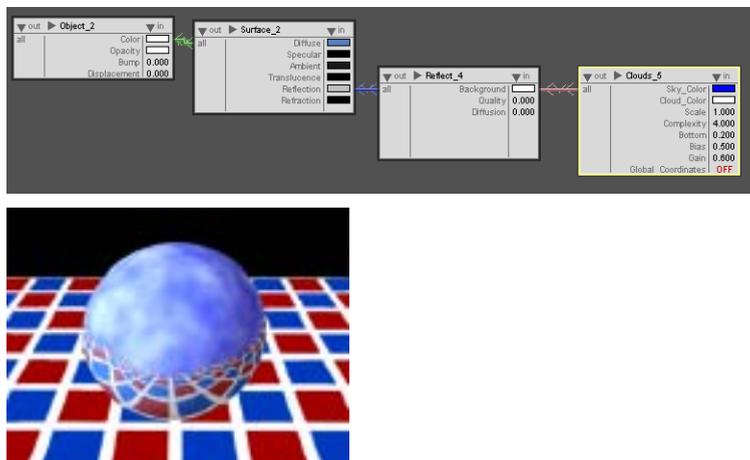
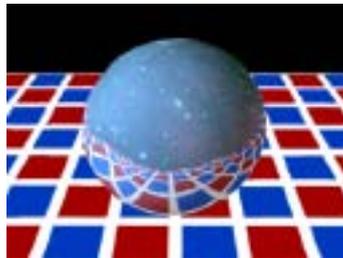
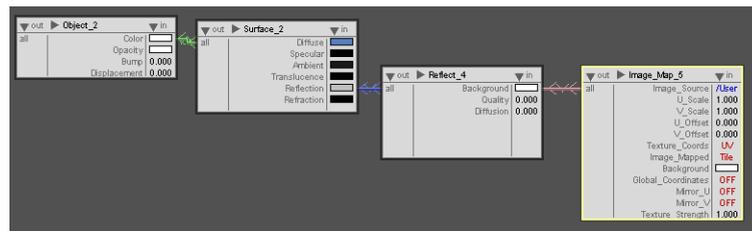


Figure 22.4: Image Map Background (stars)



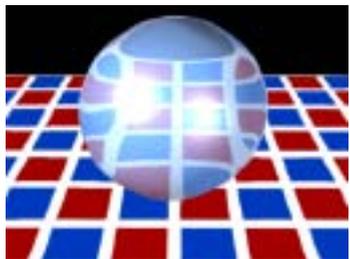
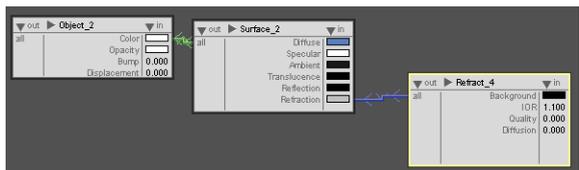
Refraction

Use raytraced refraction to get light to accurately bend when it passes through a transparent surface.

To get a raytraced transparent surface...

1. Use a Surface or SM3Surface node along with a Refract node.
2. Plug the Refract node into the Refraction input of the Surface or SM3Surface.
3. Click on the Refraction input color and set it to something other than white or set the Refraction value on the SM3Surface to something less than 1. (If the Refraction color is not changed from white, the image showing through the surface will overwhelm the surface characteristics of the object.)
4. Raise the IOR value just above one to bend the light as it passes through the surface. Higher values give more distortion.

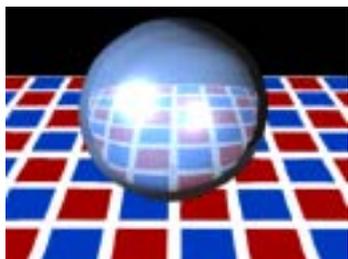
Simple Refractive Surface



Air Bubble

If you want the refractive object to be something less dense than the surrounding medium, like an air bubble in water, make the IOR less than 1.

Figure 22.5: An Air Bubble



IOR Values

For artistic reasons you may want to use more or less refraction than you find in reality. The actual IOR values of various substances are listed below.

Table 1: IOR Values

Substance	IOR
Amber	1.54

Table 1: IOR Values

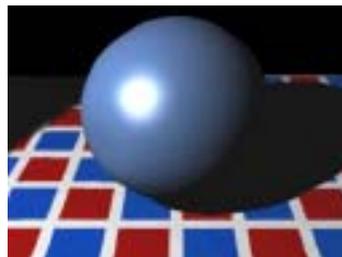
Substance	IOR
Diamond	2.45
Glass	1.44 - 1.7
Glycerin	1.47
Ice	1.3
Immersion Oil	1.5
Jade and Turquoise	1.6
Plastic	1.46 - 1.7
Quartz	1.55
Ruby and Sapphire	1.76

Raytraced Shadows

To make a light cast raytraced shadows,

In the Attribute Manager under Shadows under your light select the Shadow Type Raytraced.

Raytraced Shadow



Note: The surface receiving the shadows (in this case the tiled floor) must have some type of basic surface

Raytraced Shadows

node in its shader, otherwise the shadows will not appear.

Area Lights

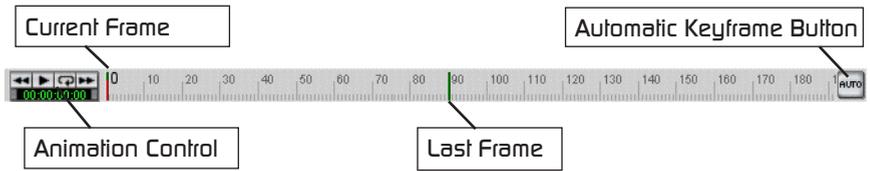
Point light sources cast crisp, hard edged shadows. Light coming from an area source creates soft shadows. The Softness setting on Raytraced Shadows makes the shadow appear as if it is cast by an area light source.

Chapter 23. Keyframes and fCurves

Timeline

The Timeline lets you control the point in time that is shown in the views.

Figure 23.1: Timeline



An animation is composed of a sequence of still images which, when displayed rapidly create the illusion of motion. In computer animation each still image is called a *frame*. When you first create a new scene the first frame, frame 0, is shown in the views. If you are creating a still image, you are typically only concerned with that initial frame.

If, however, you are creating an animation, you will work on a sequence of frames. By default, when the animation is played, each frame displays for 1/30th of a second. The number of frames displayed per second is called the *framerate*.

To change the framerate,

Go to **Render Options>Animation** in the Attribute Manager and change the entry under **fps** (frames per second).

Keyframes

One of the simplest computer animation techniques, called *keyframe* animation, works by allowing you to set up your scene as it will look at different points in time. These setups are called keyframes. The computer then generates all the intermediary frames to create a smooth transition between

your keyframes. These automatically generated images are called in-between frames or *tweens*.

Automatic Keyframes

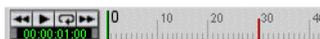
The easiest way to create keyframes is to let the system automatically do it for you.

To create keyframes...

1. Click **Auto** () in the **Timeline**. This enables auto-keyframes.
2. Adjust your scene as it should look at the start of the animation. For instance, if in your animation you are going to move a sphere, then move the sphere to the start point.

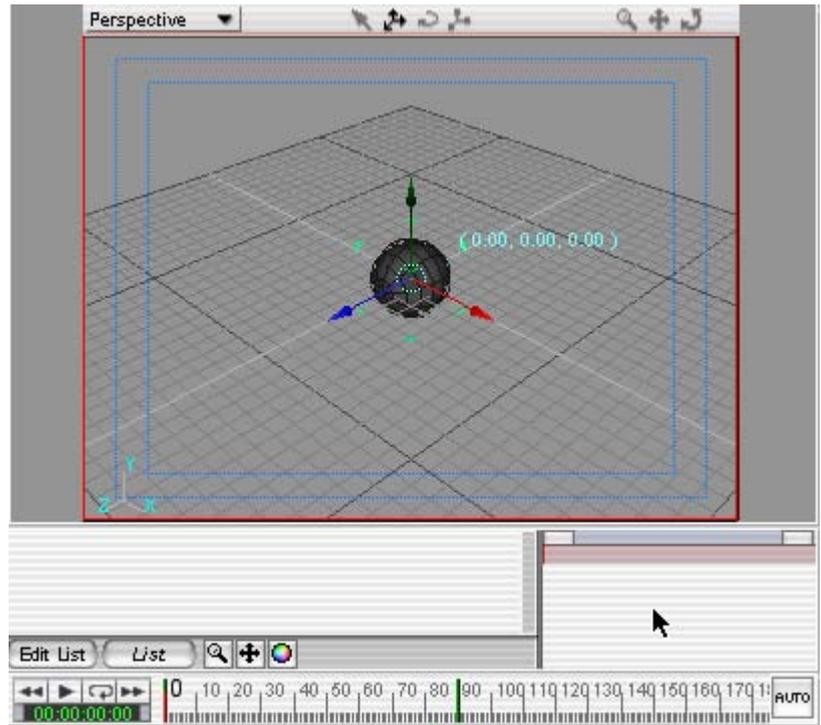
Note: You must actually adjust the parameter you are going to animate for **Auto** to actually generate a keyframe. That is, if your sphere is going to start at $0x, 0y, 0z$, where it was created, click **Auto** to enable auto keyframes, then move the sphere away from $0x, 0y, 0z$. This generates a keyframe. Then move the sphere back to $0x, 0y, 0z$. This sets the keyframe to $0x, 0y, 0z$.

3. Drag the current frame pointer along the **Timeline** to the point in time where you want the next keyframe.



4. Now adjust the scene as it should look at that point in time. This automatically creates a keyframe for each parameter you adjusted.

Keyframed Animation



(Click the picture to view the animation)

Manual Keyframes

Instead of letting the system create keyframes as you adjust parameters, it's sometimes more convenient to manually create a keyframe.

To manually create a keyframe...

1. Adjust the scene to the keyframe position.
2. For object you are going to keyframe, go to the Attribute Manager and open the object revealing its attributes.
3. For each attribute you are going to keyframe, select **Attach fCurve** from its **Animation** menu. You only need to attach an fCurve once, so if you have already keyframed the attribute, skip this step.

To open an **Animation** menu,

Click on the little arrow button (▾) next to the attribute you are going to animate.

4. For each attribute, after attaching an fCurve select **Add Keyframe** from its **Animation** menu.

fCurves

When you created a keyframe you also created a function curve or *fCurve*. An fCurve controls how quickly an attribute changes from one keyframe to the next.

To create an fCurve,

Select **Attach fCurve** from an attribute's **Animation** menu.

or

Create an auto-keyframe (see **Automatic Keyframes** on page 284).

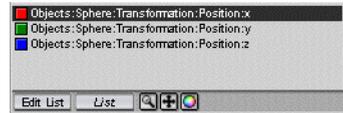
The default fCurve is S shaped. This means the change starts slowly, moves quickly halfway between the keyframes, then slows to stop at the next keyframe. This fCurve has *smooth knots*, which essentially means the transition between keyframes is smooth.

Figure 23.2: Default fCurve



To view an existing fCurve,

Select the curve from the fCurve List.



To move the view of an fCurve.,

Hold down the Space key and drag the view with the mouse.

or

Drag across the pan icon () in the fCurve widget.

To vertically scale the fCurve view,

Hold down the Option key and drag in the view with the mouse.

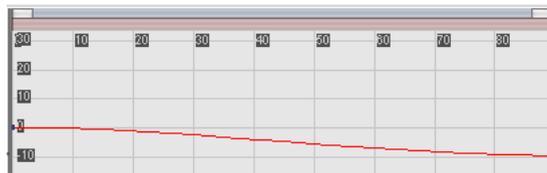
or

Drag across the zoom icon () in the fCurve widget.

To horizontally scale the fCurve view,

Drag the sliders at the top of the view left or right. By default the fCurve extends from one side of the view to the other.

fCurve Zoom Sliders



fCurve Zoom Sliders

(Click the picture to view the animation)

To edit an fCurve...

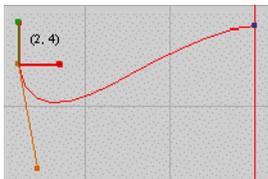
1. Select the curve from the fCurve List. This opens the curve in the fCurve editor.
2. Click on a keyframe point to select it.
3. Attached to a selected keyframe point there is an orange control handle. Drag this handle to change the shape of the fCurve. In Figure the action will start out quickly, then slow to a halt.

Edited fCurve



You can also drag the keyframe point to change the attribute start value and the point in time where the animation starts. In Figure the action starts four frames into the animation. At first the motion is backwards, toward zero, then the motion reverses and slows to a stop at about eight in frame 30.

Edited Keyframe Point



Constraints

Constraints help you control how an object moves.

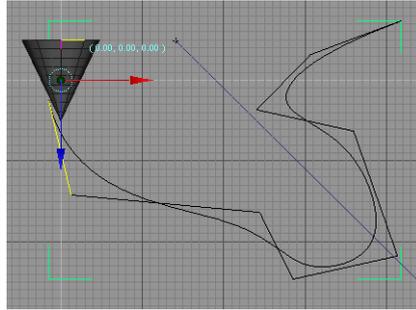
Path

Path constraints help you move an object so that it follows a specific course.

To add a path constraint...

1. Draw the path that the object will follow. You can use any of the curves on the Curves menu.

Curve for Path



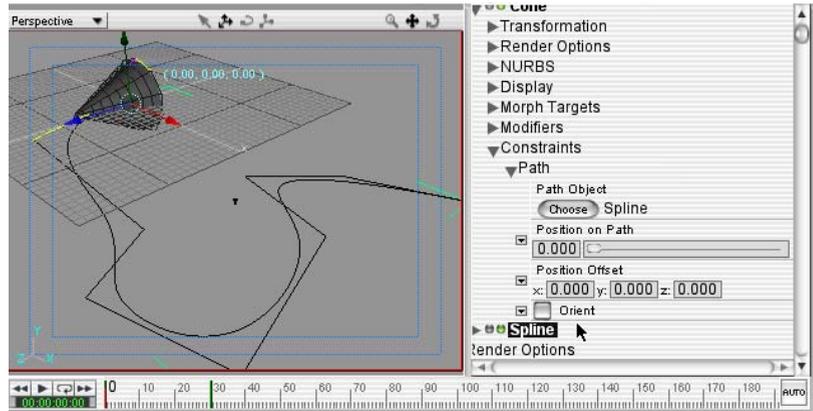
2. Select the object that will follow the path.
3. Choose **Control>Add Constraint>Path** from the main menu.
4. When prompted, select the curve you created for the path.

To animate an object along a path constraint...

1. Go to the frame where you want the animation to start.
2. Add a keyframe for the start settings under the **Path** constraint in the Attribute Manager.
3. Go to the frame where you want the animation to end.
4. Adjust the **Path** constraint settings in the Attribute Manager and add a keyframe. (You can also add as many intermediary keyframes as you like between the start and end frames.)

Constraints

Path Animation



(Click the picture to view the animation)

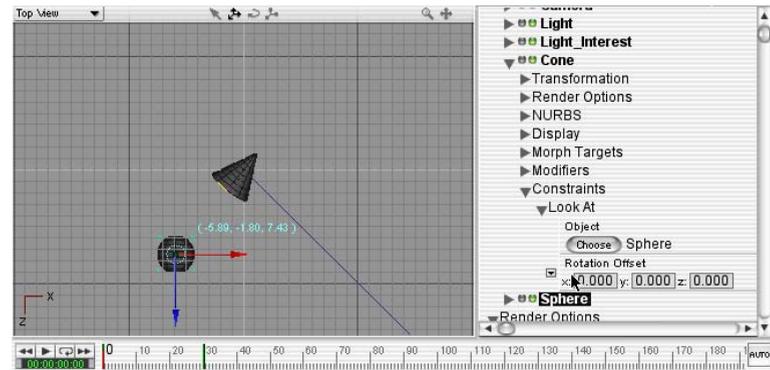
Look At

Look at constraints make an object keep one face toward another object.

To add a look at constraint...

1. Select the object that is going to look .
2. Choose **Control>Add Constraint>Look At** from the main menu.
3. When prompted, select the object that the other object is going to face.

Look At Animation



(Click the picture to view the animation)

Orbit

Orbit constraints make one object revolve around another.

To add an orbit constraint...

1. Select the object which is going to move in an orbit.
2. Choose **Control>Add Constraint>Orbit** from the main menu.
3. When prompted, select the object around which the other object is going to orbit.

Snap

Snap constraints make one object keep in sync with another object.

To add a snap constraint...

1. Select the object which is going to follow.
2. Choose **Control>Add Constraint>Object Snap** from the main menu.
3. When prompted, select the object which is going to lead.

Limits

Limits prevent attributes from exceeding specified values.

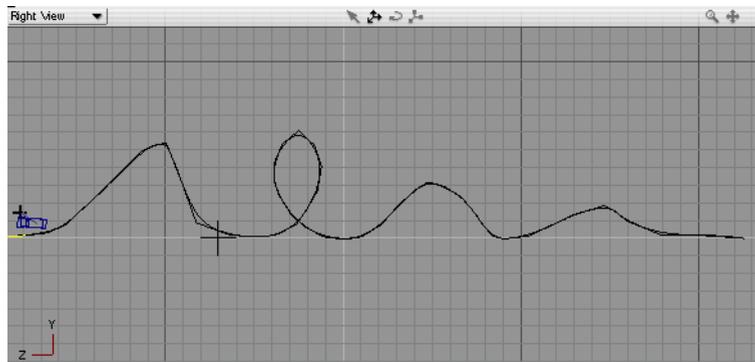
Constraints

Chapter 24. Tutorial: Roller Coaster

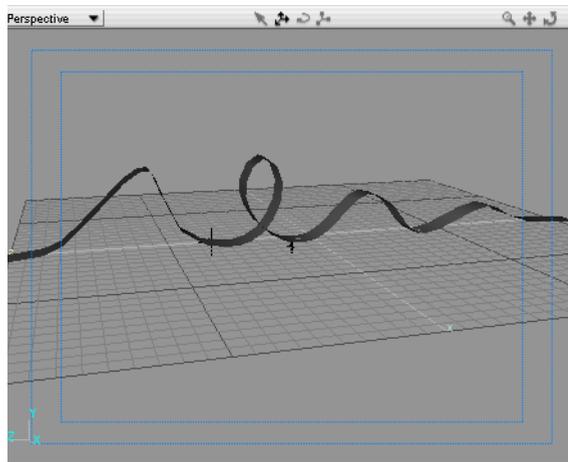
This tutorial uses the **Distance** and **Path** constraints, simple shaders, basic lighting, and animation keyframes to create a simple roller coaster animation.

First, create the track.

1. Create a spline.



2. Loft the spline to create the track.

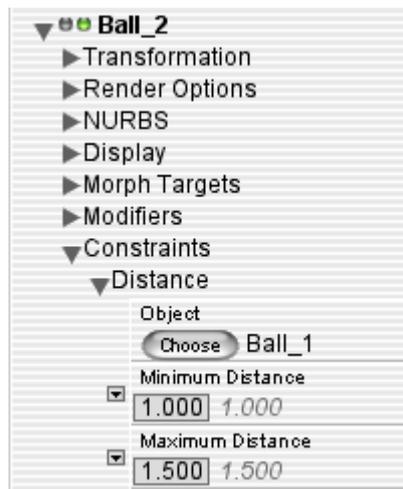


3. Create a sphere.

4. Move the sphere to the start of the track.
5. Create a second sphere by selecting **Edit > Duplicate**.
6. Move the second sphere so you can select one individually.

Add the **Distance** constraint.

1. Select the *second* sphere.
2. Select **Control>Add Constraint>Distance**. PiXELS 3D will beep and the message "Select Target Object" will appear on the status bar.
3. Select the *rst* sphere.
4. In the Attribute Manager, the **Distance** constraint is displayed.



Note: The *rst* sphere in this example was named **Ball_1**.

5. In the **Minimum Distance** field, enter the radius of the *rst* sphere. This will keep the second sphere from overlapping it.

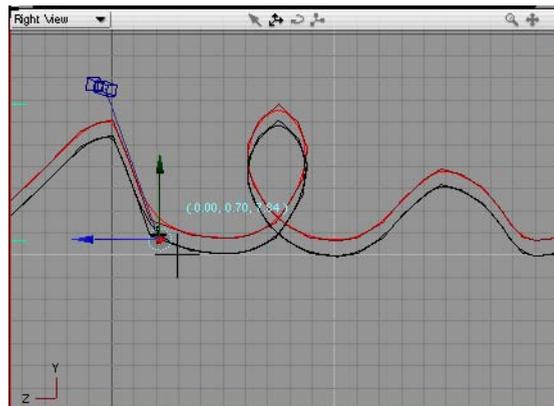
6. Enter another value in the **Maximum Distance** field to keep the second sphere near the first.

Build the train by creating and linking more spheres.

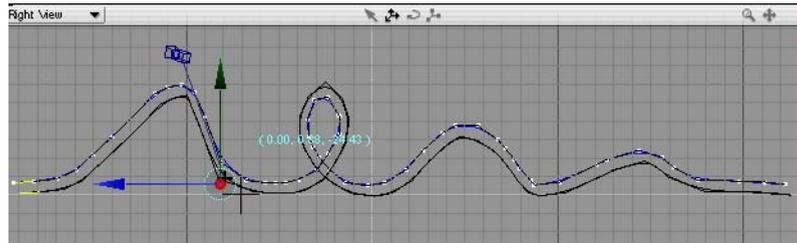
1. Select only the second sphere.
2. **Duplicate** the sphere by selecting **Edit > Duplicate**. Note that the **Distance** constraint is also duplicated. However, it points at the first sphere.
3. Select the third sphere.
4. In the Attribute Manager, click the **Choose** button under the **Distance** constraint.
5. Select the second sphere.
6. Repeat the above steps, linking each sphere to the one before it, until you have the desired number of spheres.

Next, create a path for the train to follow.

1. Select the spline you used to create the track.
2. **Duplicate** the spline. The new spline will be the path.
3. Move the path so that it is above the track. This illustration shows the path in red.

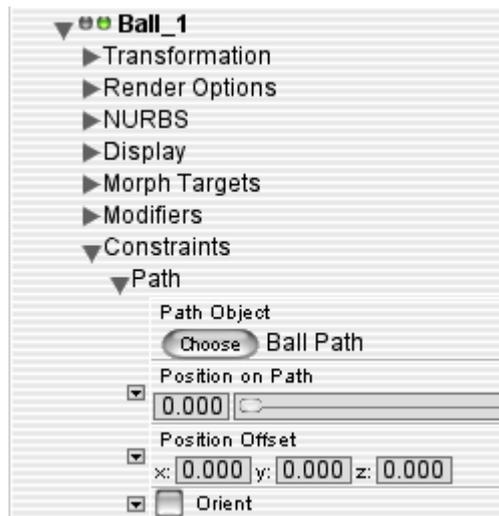


4. If there are places where the path spline goes "under" the track, select vertices and move them until the entire path is on the same side of the track. If you include a loop, for example, you will need to edit your path so it stays on the correct side of the loop.



Make the train follow the path.

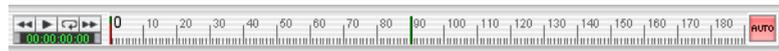
1. Select the first sphere.
2. Select **Control>Add Constraint>Path**.
3. Select the path created earlier.
4. In the Attribute Manager, the **Path** constraint is displayed.



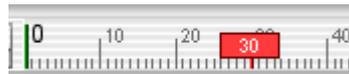
5. Drag the slider on the **Position on Path** slider bar to move the train of spheres.

Next, set up the motion that is going to occur during the animation.

1. Move the spheres away from the start of the path.
2. Click the **AUTO** button in the timeline bar to turn on automatic keyframe generation.
3. Move the spheres to the start of the path. This will automatically create a keyframe at frame zero.



4. Click in the timeline to specify the last frame number. You can drag the handle to a position anywhere on the timeline.

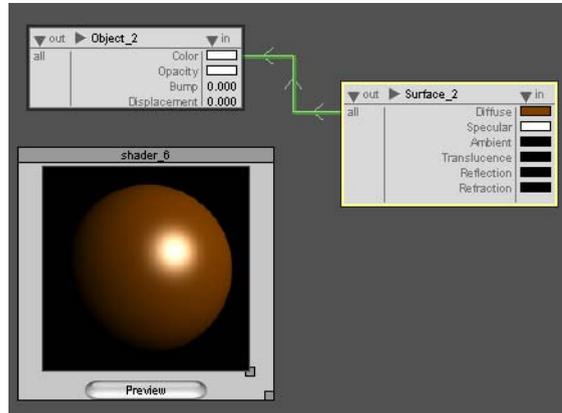


5. Use the **Position on Path** slider to move the first ball to the end of the path.
6. Click the **AUTO** button in the timeline bar to turn off automatic keyframe generation.

Next, add a shader to the track.

1. Switch to the ShaderMaker layout.
2. In the Attribute Manager, select the track.
3. In the Shader Manager, click the **New Shader** button.
4. In the ShaderMaker window, click the **Nodes** button and select **basics > surface**.
5. On the **Surface** node, click the **Diffuse** field to choose a base color for the track.

6. On the **Surface** node, click the **Specular** field to choose a highlight color for the track.
7. Drag your cursor from the **all** entry under **out** on the **Surface** node to the **Color** input of the **Object** node.
8. Click the **Preview** button to see what the shader will look like.

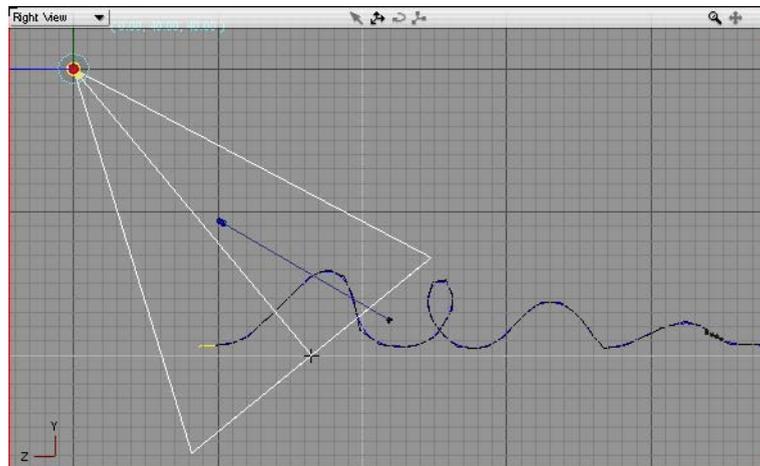


Use the same method to add shaders to the balls in the train.

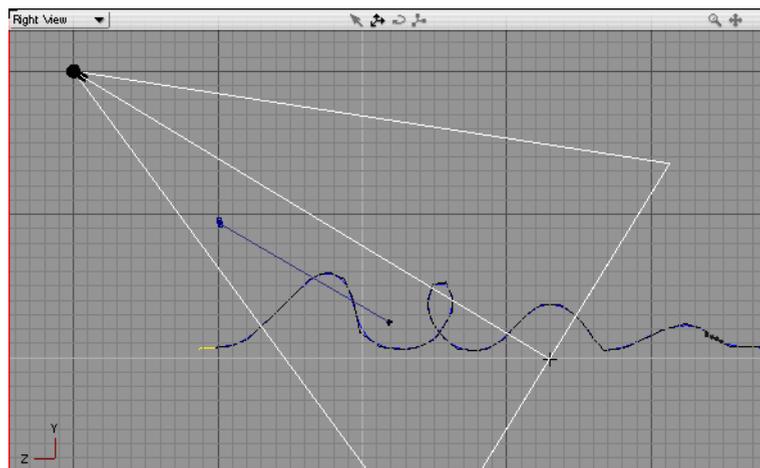
Next, set up lighting for the animation.

1. In the Attribute Manager, select the **Light** and expand its option list.

2. In one of the non-perspective views, zoom out until you can see the light cone.



3. Click the **Always Show Cone** option box.
4. Select the **Light_Interest**.
5. Move the **Light_Interest** so that the light cone includes the main interest point of your animation.



Add a second light for fill-in illumination.

1. Select **Special>Light**.
2. In the **Light_1** options, select **Always Show Cone**.
3. Change the **Light Type** to **omni**.
4. Change the **Intensity** to **0.600**.

Now, set up the render options for the animation.

1. Select the **Perspective View**.
2. Click the word **Perspective** to open the view's drop-down menu and select **Expand to Full View**.
3. Adjust the magnification and camera angle as desired.
4. In the **Attribute Manager**, click the triangle to the left of **Render Options** to expand the options list.
5. Expand the **Image Setup** options list.
6. Select a size for the animation, and enter the desired size values, in pixels, in the **Width** and **Height** fields.
7. Expand the **Animation** options list.
8. Check **Render Animation**.
9. Change the **End Frame** value to the last frame number.

Select a **Render** command from the **File** menu.

To turn your rendered frames into an animation, you can use **Apple Quicktime Pro**, or you can use **Flipbook**, which comes with **PiXELS 3D**. For details on using **Flipbook**, see **Animations** on page 246.



Chapter 25. Particle Animation

Particle Systems

Particle systems are used to model and animate things like bubbles, liquids, smoke, or anything else where there are lots of little parts that move in roughly the same manner.

To create a particle system,

Select **Particles>New Particle System** from the main menu. This creates a particle system with one emitter.

PiXELS creates true particle systems. That is, each particle is a 3D object which is rendered along with the rest of the scene. Some 3D software packages create particle systems as a post-production effect. That is, the particles are rendered and pasted onto the image after the rest of the scene is rendered. That causes the particles to look at or display artifacts. The PiXELS approach of creating systems of actual 3D objects gives much more realistic and satisfying results.

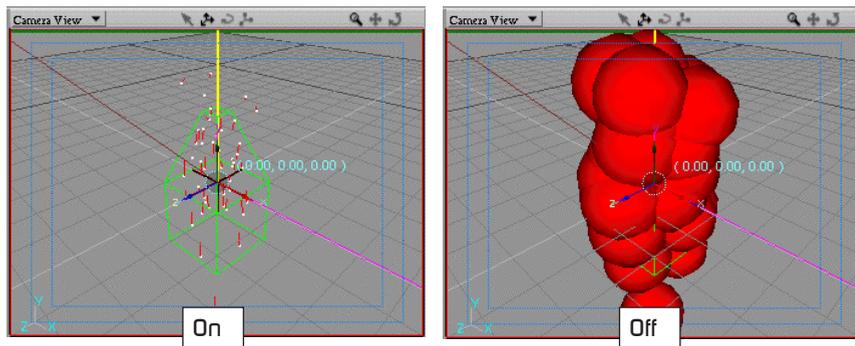
Emitters

An *emitter* creates particles. The attributes of an emitter determine how the particles look.

Display and Appearance

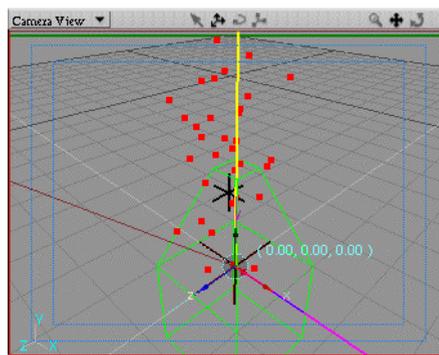
Only Show Motion. With this checked, each particle is shown on screen as a point with a line attached. The line trails from the particle like a kite tail. Its length shows the speed of the particle and its direction shows the particle's direction of motion.

Figure 25.1 : System with and without Only Show Motion



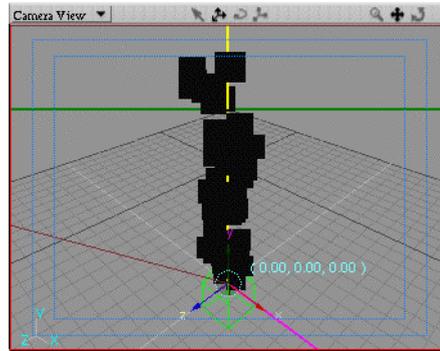
Particle Type.

Points. Basic dots. Points render quickly but are usually not as interesting as the other types.

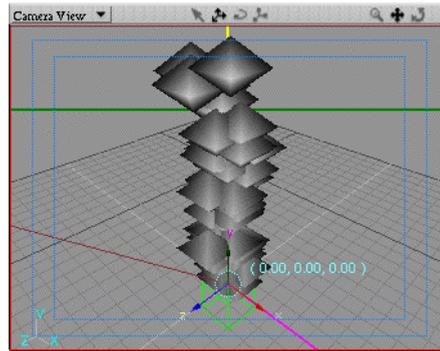


Screen Square. A screen square is a two dimensional square that always faces the camera. Screen squares render very quickly, so if you have a complex shape you want to animate as a particle system (say a school of jelly-

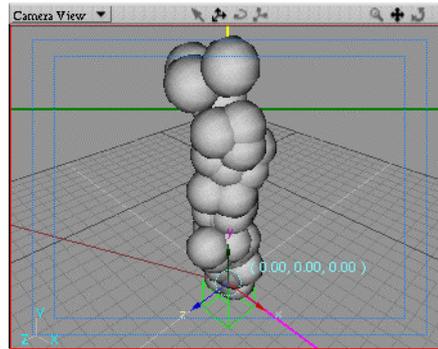
sh), you can make an image map of the shape and map it to the screen squares.



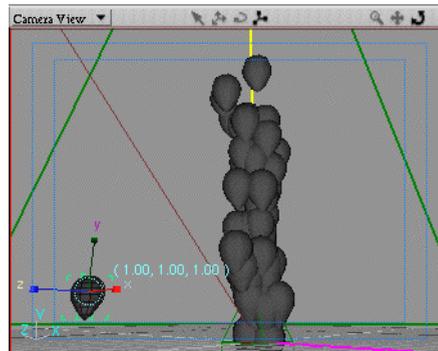
Octahedron. Eight sided solids with triangular faces. These do not render as fast as screen squares, but render quickly in comparison to the other types.



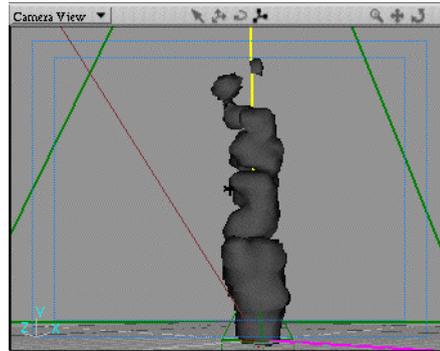
Sphere. Spheres render even slower than tetrahedrons.



Instance. Uses any object you want (even an object that has other objects nested under it) as the particle. Type the name of the object in the **Instance Options>Object** field. The more complex your object, the more time the scene will take to render.



Blobby. Animates a metaball system with each particle as a separate blobby object.



Life and Death

Particles are created from the **Emitter**. Each particle moves according to the **Forces** and **Dynamics** applied to it. Then, after a certain number of frames each particle is removed from the scene (dies).

Max Particles. As long as more particles are created than destroyed, the number of particles increases. If the number becomes equal to or greater than the **Max Particles**, no more are created until the number falls back below this number.

Emit From. The geometry of the volume in which particles are born.

Point. Particles are born in a spherical volume around a point.

Path. Particles are born in a volume around a curved path. Specify the name of the curve as the **Emission Object**.

Surface. Particles are born in a volume around a surface. Specify the name of the NURBS surface as the **Emission Object**.

Particles Travel. The starting direction of particle velocity. **Uni-Directional** particles start out traveling in approximately the same direction. **Omni-Directional** particles start out traveling in random directions.

To change the direction of **Uni-Directional** particles,
Rotate the particle **Emitter**.

Emission Radius. The radius of the volume around the **Emission Object**. Particles are born in this volume.

Birth Rate (Editor). The number of particles that appear on-screen in the views per frame.

Birth Rate Multiplier (Render). The number of particles that actually get rendered for each particle on-screen. If the multiplier is 2 and the number of particles on-screen is 25, then 50 particles get rendered.

Velocity. The initial speed or magnitude of the particle velocity. The higher the number, the faster the particles move when they are created.

Lifespan. The average number of frames that each particle will survive.

Deviation

Usually, when you use particles, you want a random cloud of particles that all follow the same rules.

Dynamics

Dynamics govern how particle motion changes over time. Dynamics can have different sources.

Local Dynamics. These are the Attribute Manager settings nested under **Dynamics** under the particle system. **Local**

Dynamics can only affect particles within a single particle system.

Scene Dynamics. These are the Attribute Manager settings under **Scene Parameters>Dynamics**. Scene Dynamics can affect some or all of your particle systems.

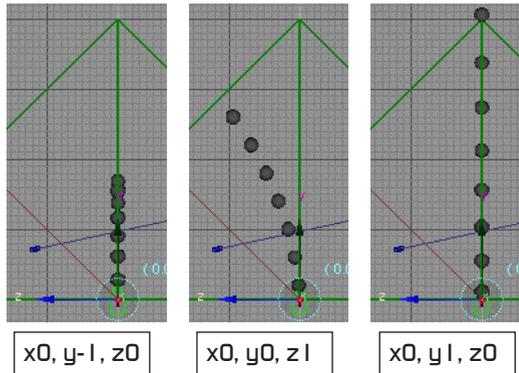
Only User Forces. This turns off any effect specified under **Dynamics**. Only Forces you apply to the system will affect it. Forces are like dynamics modifiers applied to your particle system.

Gravity

Gravity accelerates or increases particle velocity in one direction. If the particle's initial velocity is down and the gravity direction is down, the particle will fall faster and faster. If the particle's initial velocity was up, and the gravity direction is down, the particle will move up, slow, then fall at an increasing speed.

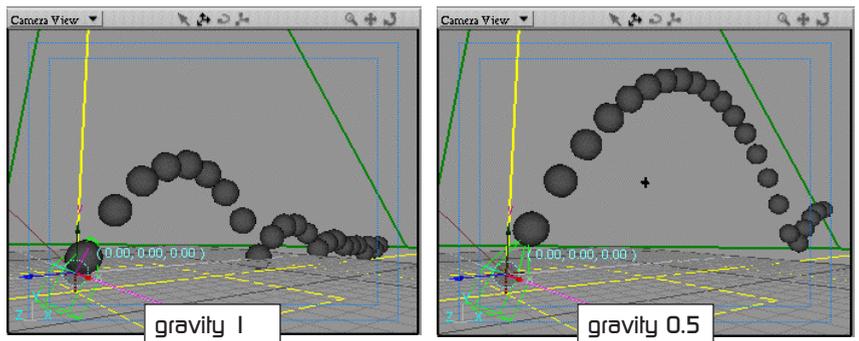
Gravity Direction. The direction in which gravity pulls each particle. The particles in Figure 25.2 are all aimed up from a small point source with very little deviation at a rate of 1 every 4 frames (**Birth Rate** 0.25). In all 3 pictures the animation has run for 30 frames. In the picture at the left, the gravity is down (x0, y-1, z0). In the middle picture, the gravity is to the side (x0, y0, z1). In right picture, the gravity is up (x0, y1, z0).

Figure 25.2: Gravity Direction



Gravity Strength. The amount of change in velocity. The particles in Figure 25.3 are all aimed at an angle (rotation -25) from a small, point source with very little deviation at a rate of 1 every 4 frames (**Birth Rate 0.25**). In both pictures the animation has run for 90 frames. In the picture on the left, the gravity is 1. In the picture on the right, the gravity is 0.5.

Figure 25.3: Gravity Strength



Wind

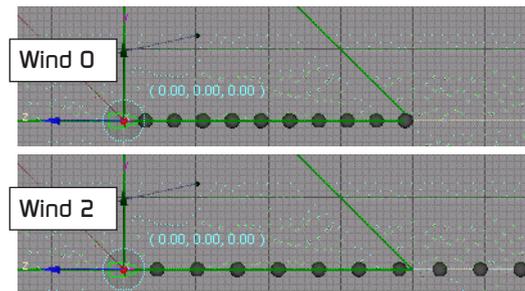
Wind carries particles along with it. The wind motion is added to the particle motion.

Wind Direction. The direction in which the wind pushes the particles. Specify a position. A line drawn from x_0, y_0, z_0

through that position points in the same direction as the wind.

Wind Strength. The amount of movement that the wind adds to the particles. **Wind Strength** is scaled by 5, so a particle moving a a velocity of 1 against a wind of 5 would be standing still.

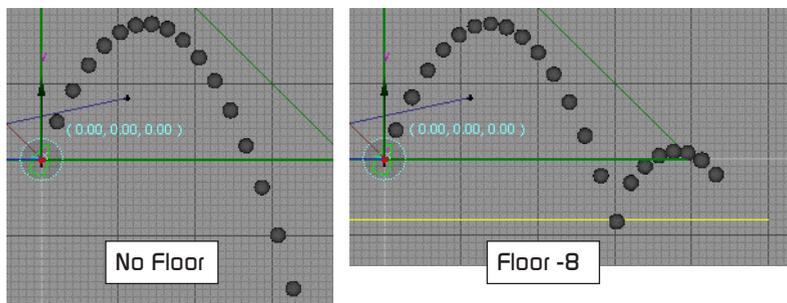
Figure 25.4: Wind Strength



Floor

If a particle hits the floor, it bounces. If a particle is created below the floor, it is moved to the floor surface.

Figure 25.5: Floor



Forces

Forces act on particles to move them around, much like **Dynamics**. A **Force**, however, is applied to a particle system in much the same way a **Modifier** is applied to an object. You can

Forces

stack multiple **Forces** of the same or different types to create complex particle motions.

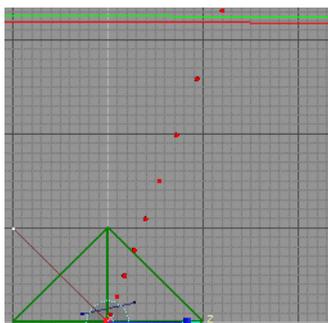
All **Forces** have a bounding box or **Shape** within which the force applies to particles. When particles leave the boundary, the force no longer applies. Boundaries are either **Box** shaped or **Spherical**. Boundaries can be moved, scaled, or rotated just like objects.

Like **Modifiers**, all **Forces** have an **Active** checkbox that can be used to temporarily turn the force on or off.

Accelerator

Accelerates a particle's speed in its current direction of motion.

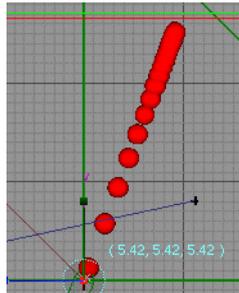
Figure 25.6: Accelerator Force



Friction

Reduces a particle's speed in its current direction of motion until the particle comes to a stop.

Figure 25.7: Friction Force

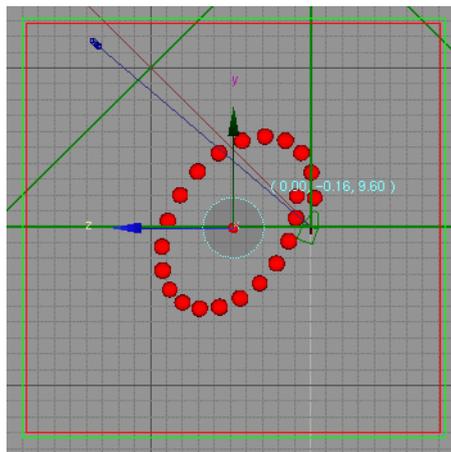


Gravity

Like the Gravity under Dynamics except its effect is restricted to the bounding Shape.

Gravity Well. When this is enabled, the gravity pulls toward the center of the bounding Shape.

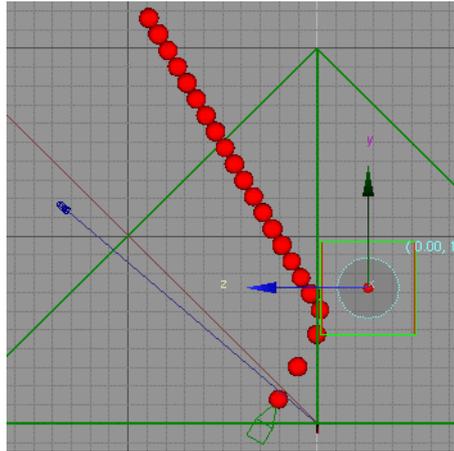
Figure 25.8: Gravity Well



Collision

Reflects particle motion as if the bounding Shape were a solid object.

Figure 25.9: Collision Force

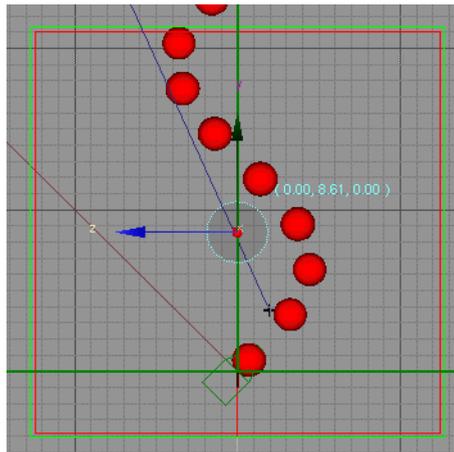


Invert. When enabled, this reflects particles backwards rather than at the usual angle of reflection.

Vortex

Moves the particles in an orbit around the Vortex Axis.

Figure 25.10: Vortex Force



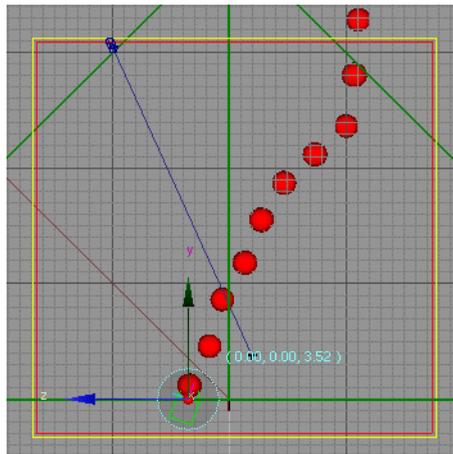
[Click the picture to view the animation]

Affect Velocity. Restricts particle velocity to a plane perpendicular to the **Vortex Axis**.

Turbulence

Moves the particles in random directions. A noise function is used to determine the direction and magnitude of the push in any given region of the bounding **Shape**.

Figure 25.11: Turbulence Force

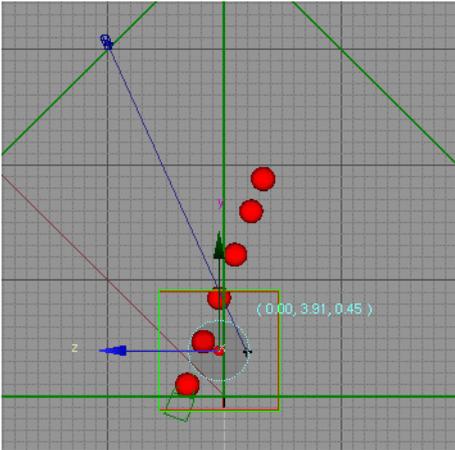


Noise Scale. Scales the region over which the pushing direction due to the turbulence changes.

Death

Ages the particles by the specified amount per frame.

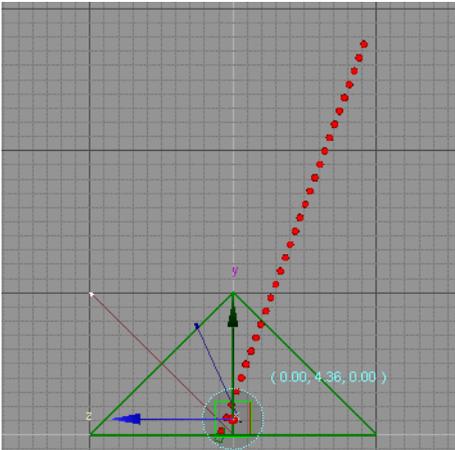
Figure 25.1 2: Death Force



Life

Increases the particle lifespan by the specified amount per frame.

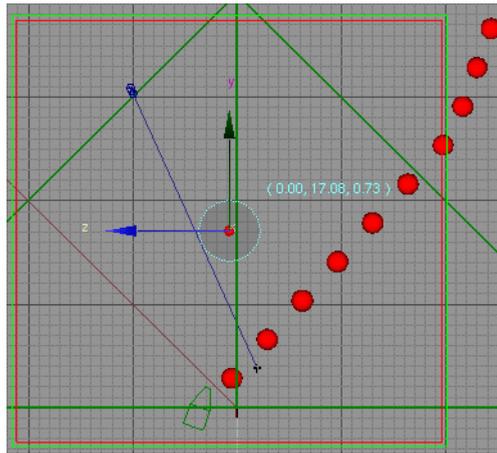
Figure 25.1 3: Life Force



Wind

Like the Wind under Dynamics, this pushes the particles in the wind direction.

Figure 25.14: Wind Force



Chapter 26. Character Animation Tools

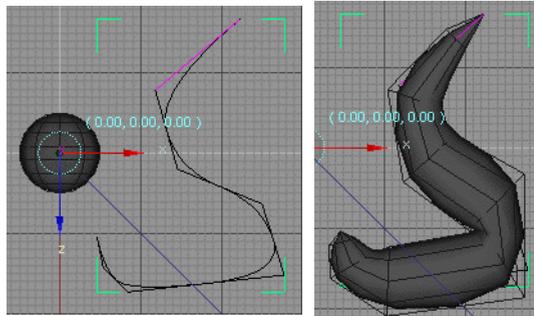
Spline Deformers

A spline deformer stretches an object along a curve.

To use a spline deformer...

1. Create the object you want to deform.
2. Create the spline curve.
3. Choose the object and select **Edit NURBS>Add Modifier>Spline Deformer**, then select the spline when prompted. This stretches the object along the entire length of the spline.

Figure 26.1: Spline Deformed Sphere



Note: When you apply a **Spline Deformer** it may at first look like the deformation went haywire. If this is the case, check **Modifiers>Spline>Axis** under your object in the Attribute Manager to make sure the spline is deforming your object along the intended axis. Typically you want to use an axis parallel or tangential to v.

Try playing with the **Scale** and **Offset** attributes in **Modifiers>Spline** (under the object in the Attribute Manager) and watch your object squish along the spline.

Note: In general, Deformers like the Spline Deformer, Lattice Deformer, and Bone Deformer should be applied before Transformations (Move, Scale, or Rotate).

If you do Scale or Rotate the object before adding a Deformer, Park the object first (select Control>Park from the main menu). This resets the object so the current scale is considered to be 1 and the current rotation is considered to be 0.

To Deform and Move an object, take note of the position where you want the object, move the object to x0,y0,z0, apply the Deformer, and move the Deformer to the noted position.

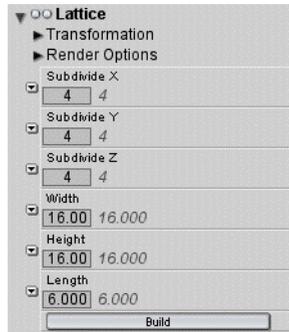
Lattices

A *lattice* is a framework or scaffolding you can put around an object to control the way it deforms.

To use a lattice...

1. Create the object you want to deform.
2. Choose Special>New Lattice from the main menu. This creates an editable lattice.
3. Use the editable parameters to scale the lattice or change the number of subdivisions so that it encloses the part of

the object you want to be able to deform (usually the whole object).



Note: The editable parameters are **Subdivide X, Y, and Z, Width, Height, and Length**. Do not **Transform** the lattice (**Move, Scale, or Rotate** it) or it will not predictably deform your object.

4. In the Attribute Manager under the lattice, click **Build**. This sets the start position of the lattice.
5. In the Attribute Manager, make sure the **Transformations** of the object you want to deform are set to 0 and drag the object so it is nested under the lattice.
6. Select the object you want to deform and choose **Edit NURBS>Add Modifier>Lattice Deformer**. When prompted, select the lattice. This adds the lattice as a **Modifier** to the object.

Now when you move a lattice vertex it deforms the part of the object contained in that lattice cell (assuming the **Lattice Modifier** under the object in the Attribute Manager is **Active**). When you **Move, Scale, or Rotate** the lattice itself the transformations are also applied to the object because the object is a sub-object of the lattice.

Figure 26.2: Lattice Modifier



Note: As noted on page 320, Deformers like the Spline Deformer, Lattice Deformer, and Bone Deformer should generally be applied before Transformations (Move, Scale, or Rotate).

Bones and IK

An *Inverse Kinematic (IK)* system is used to animate a bendable object like an animal or human. In PiXELS, IK systems are composed of *bones* which govern the bending of the objects to which they are attached. If you move one end of an IK system, for example the bone at the tip of a finger, all the connected bones follow in a natural motion. The movement of the bones, in turn bends the body to which they are connected.

You can use an IK system by either attaching objects directly to the bones in the system or by applying the system as a bone deformer to one or more objects.

The first method moves each object with the bone to which it is attached, but does not distort the object. This is perfect for making a robot move or for animating characters in situations where seams at their joints will not be noticeable.

Prepare Your Model

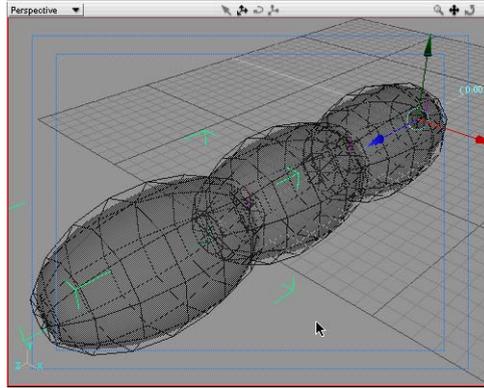
First you need to prepare your model.

To prepare your model...

1. Create the objects which are going to bend relative to each other.

2. Park each object individually by selecting **Control>Park**.

Model Ready for Bones



3. To make your objects look transparent, use **ctrl-click** and select **Stipple Surfaces**.

Create the IK System

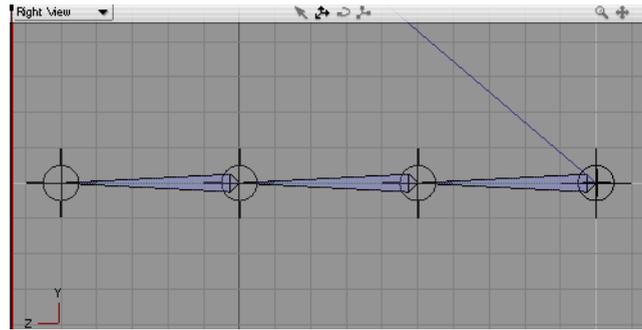
1. From the main menu select **Special>Inverse Kinematics>IK Chain**. This opens the IK Chain dialog.



2. Change the Number of Segments to the number of objects you have and click **OK**. In the Attribute Manager, you should see the IK Chain root:



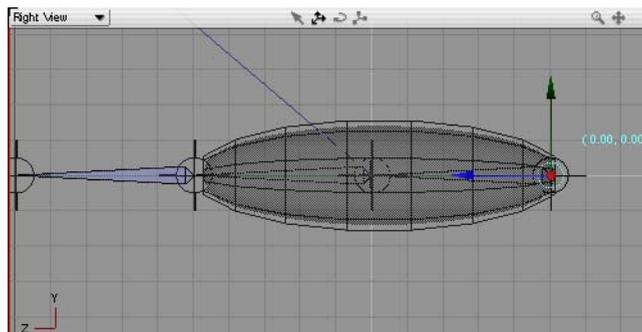
3. Click the **Set Rest Orientations** button.
4. Hide your objects by clicking on the green dots immediately to the left of the objects' names. You should see something like this:



Position the IK System

You will need to move the IK so the first bone starts where the first object starts. The start of a bone is the wide end of the bone, like the start of the arm is at the shoulder.

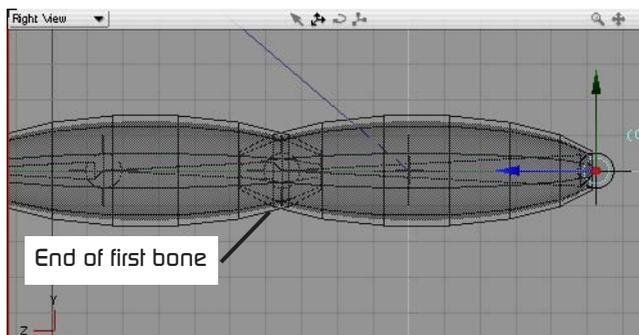
1. Unhide your first object by clicking on the red dot next to its name.
2. Click on **IK Chain root** to select the whole bone system.
3. Move the wide end of the bone system to the start of your first object.



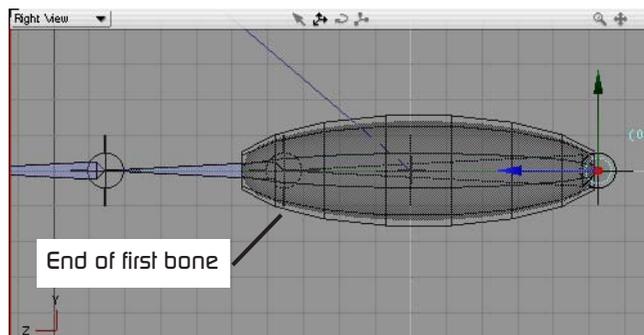
4. Click **Set Rest Orientation**.

Next, you need to adjust the length of the first bone.

1. With the object visible, use the Attribute Manager to select the IK Chain root.
2. Expand the **Bone** attribute.
3. Change the **Bone Length** value until the narrow end of the bone coincides with the center of the first joint.



Note: This can be difficult to see, so show and hide your objects as necessary.

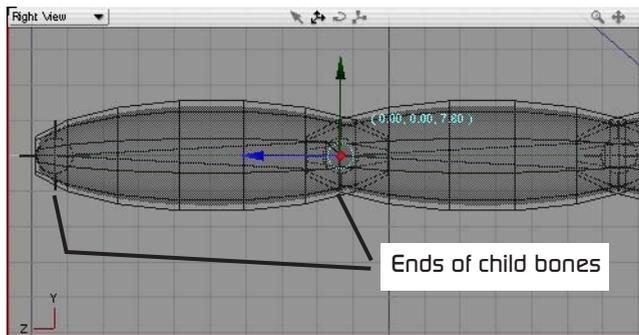


4. Once the bone is oriented in the correct rest position, click **Set Rest Orientation**.

Next, adjust the length of the other bones.

Note that the last item in the **Bone** attribute is another **Bone** attribute. This second bone is a *child* bone, dependent on the first bone.

1. Expand the attributes of the first child bone.
2. Adjust the **Bone Length** as you did with the first bone.
3. Repeat with the second child bone.

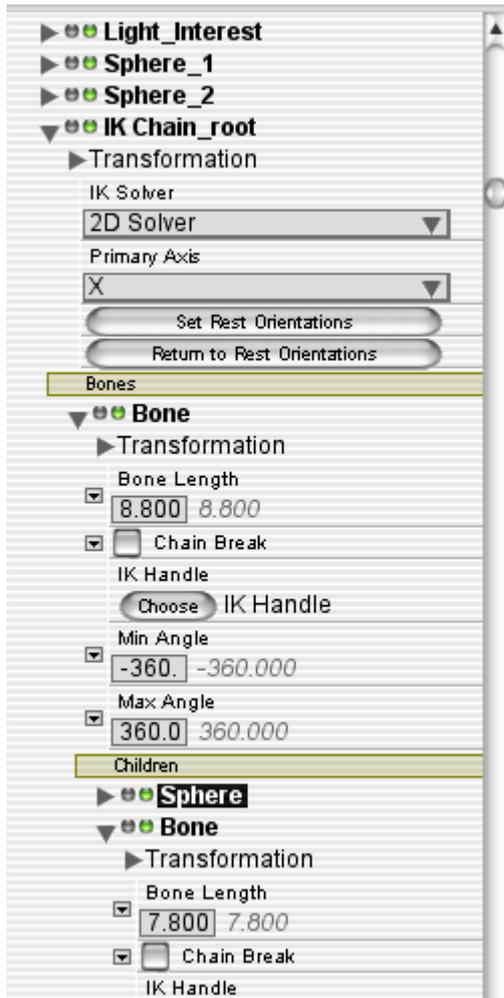


4. Click **Set Rest Orientation** when you have everything positioned.

Attach the Bones

1. Drag the first object under the first bone in the Attribute Manager.

First Bone Attached

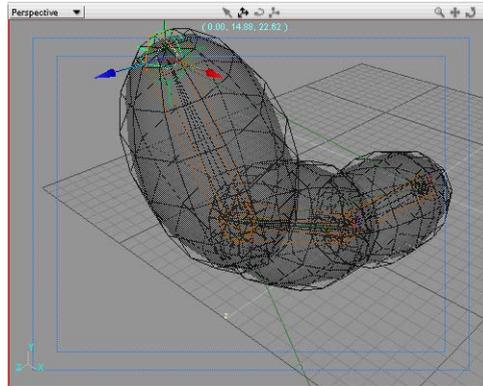


2. Drag subsequent objects under the corresponding bones in the Attribute Manager. (Note that it is best to drag the last object until its black line is under the word **Bone** to attach it.)
3. Click **Set Rest Orientation**.

Animate the System

To animate the system, move the last IK Handle in the chain. If you have attached your objects, they will follow the bones as you move them around.

Figure 26.3: Move the IK Handle

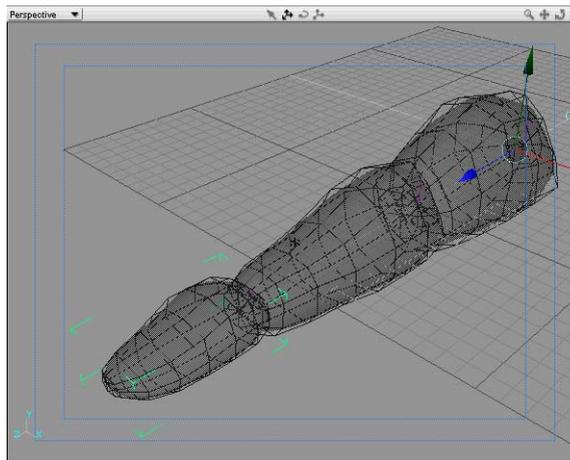


IK Modifiers

IK Modifiers are used to bend objects along with the bone system.

First you need to create and position your model. This is explained in [Prepare Your Model](#) on page 322.

Model Ready for Bone Modifiers



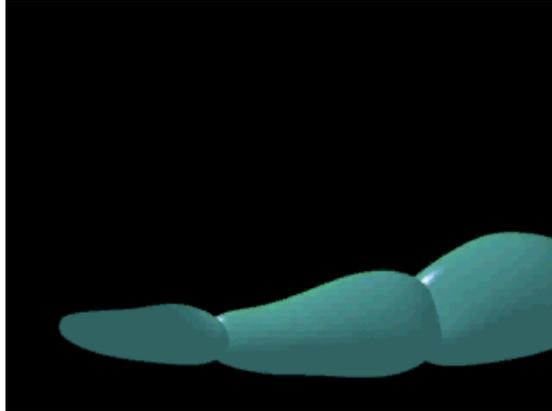
Next you need to create and position the IK system, as explained in the sections [Create the IK System](#) on page 324 and [Position the IK System](#) on page 325.

Add Bone Modifiers

1. Select the first object element and choose **Edit NURBS>Add Modifier>Bone Deformer** from the main menu.
2. When prompted, select the **IK ROOT**.
3. Repeat these steps for each object element.
4. Click **Set Rest Orientation**.

Animate the System

To animate the system, move the last IK Handle in the chain. If you have added all the **Bones** modifiers, the objects will bend in response to the bone movement.



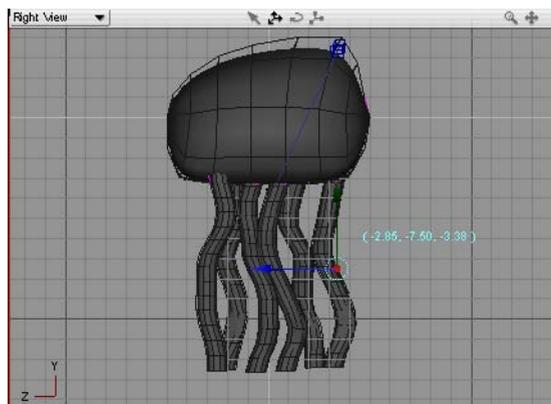
(Click the picture to view the animation)

Morph Targets

Morph targets allow you to mix different gestures or deformations in various proportions.

To set up morph targets...

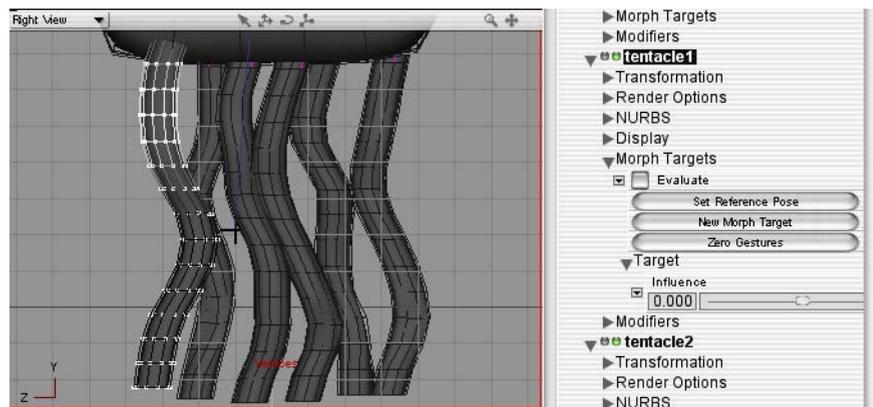
1. Create the shape you are going to morph.



2. When you have the object in a neutral shape, click **Set Reference Pose** under **Morph Targets** in the Attribute Manager.



3. Move the vertices to one of the gestures and click **New Morph Target**. This creates a morph target slider.



4. Click **Zero Gestures**, then click the **Evaluate** checkbox to enable evaluation of the morph targets. This moves the vertices back to the reference pose.
5. Disable evaluation and repeat Step 1 through Step 4 for the next gesture.

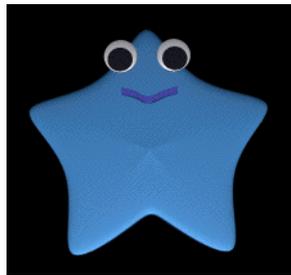
Once you have multiple morph targets created, you can use the **Influence** sliders to see how they interact.

You can also add morph targets to a system with bone modifiers. This can be used to model muscles in an arm, for example.

Lofted Blends

In character animation, your character typically has more than one limb. Humanoids, for instance, with four limbs and a head are shaped like a five pointed star.

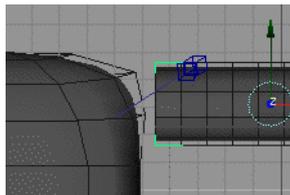
Figure 26.4: Humanoid Star



This is difficult to seamlessly represent with NURBS. The solution is to create the character with multiple NURBS objects and create blends between the objects to hide the seams. One way to create a nearly seamless blend is to wrap and project curves onto the two surfaces to be blended, then loft the projected curves together.

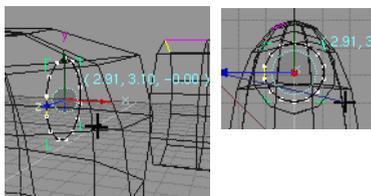
To create a lofted blend...

1. Create the two surfaces to be blended.

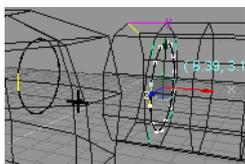


2. Create a Circle and select **Edit NURBS>Make Editable** to make it into an editable NURBS curve.

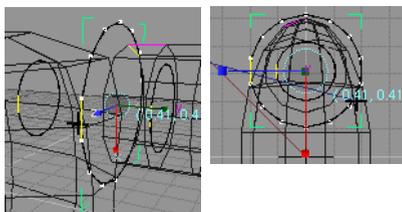
3. Move, rotate, and scale the NURBS circle so that it is inside one of the objects.



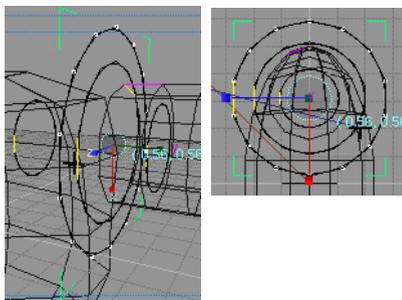
4. Duplicate the NURBS circle and move the duplicate so that it is within the second object.



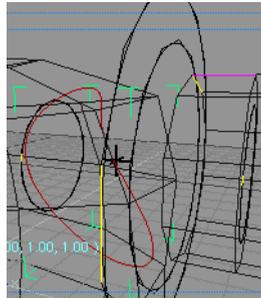
5. Duplicate the NURBS circle again, move it between the two objects and make it larger than parts to be blended.



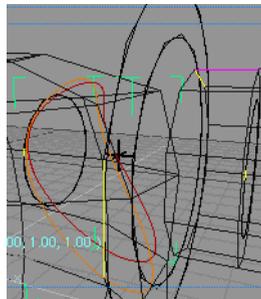
6. Duplicate the NURBS circle again and make it even larger, keeping it the same distance between the two objects.



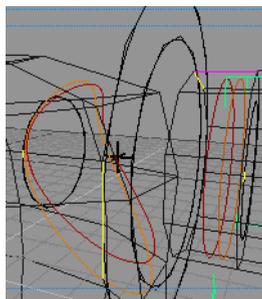
7. From the main menu choose **Edit NURBS>Wrap and Project**. When prompted for the first curve, select one of the small circles. When prompted for the second curve, select one of the larger circles. When prompted for the surface, select the surface that contains the small circle. This produces a curve wrapped to the surface.



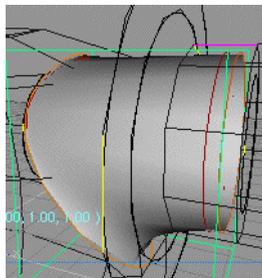
8. Repeat the previous step, except select the other larger circle when prompted for the second curve. This produces another curve wrapped to the surface.



9. Repeat Step 7 and Step 8 for the other small circle and curved surface.

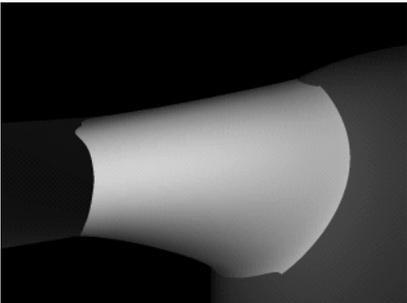


10. Create a loft from the outside curve on one object through the inside curve on the same object, through the inside curve on the other object, to the outside curve on the other object.



This blend currently does produce a bit of a seam in the rendering (the seam can easily be hidden with a bumpy shader). It does, however, animate well since the curves move with the objects they are projected onto and the loft moves with the curves.

Figure 26.5: Lofted Blend



Chapter 27. PixelScript

PixelScript is a programming language that you can use to control your animations or add capabilities to PiXELS. This language is based on a scripting language called Tcl (pronounced tickle). PiXELS has a built-in Tcl interpreter for standard Tcl commands and custom extensions for PiXELS specific commands.

Learning Tcl

There are a wide variety of resources available for you to learn Tcl, including:

Books. The following books provide a good basis for learning Tcl:

Teach Yourself Tcl/Tk in 24 hours: ISBN: 0-672-31749-4

Tcl/Tk in a Nutshell: ISBN: 1-56592-433-9

Tcl/Tk for Real Programmers: ISBN 0-12-261205-1

Tcl/Tk for Dummies: ISBN: 0-7645-0152-6

Tcl/Tk from the Ground Up: ISBN 0-07-212106-8

Practical Programming in TCL and TK: by Brent B. Welch

On-line. The following are just a few of the many sites offering Tcl resources:

www.scriptics.com

tcl.foundries.sourceforge.net

For more information about writing plug-ins and scripts for PiXELS 3D go to www.pixels.net/developer

Creating/ Editing Scripts

To create a script, use PSEdit or your favorite text editor (such as BB Edit).

To run your script from Terminal,

Click Go.

To run a script created in a text editor,

1. Exit from PiXELS.
2. Place the completed file in the PiXELS3D 4 folder in the resources/menus/Scripts subfolder.
3. Launch PiXELS.
4. Select your script from the Scripts menu.

Expressions

Expressions are a way to programatically animate things in your scene.

To create a new expression...

1. Click on the animation menu button () next to the attribute you want to animate.
2. From the resulting menu select Attach Expression.
3. This opens the PixelScript editor, PSEdit, with a simple script that sets the current attribute to 0.
4. Using PixelScript write a mini program to animate the attribute.
5. Save the tcl file.
6. If you want on-screen feedback to show you how your expression works, enable Constant Script Evaluation in the Attribute Manager under Preferences/Scripting.
7. When you are satisfied with the results, render your animation.

Menu Commands

All menu items in PiXELS work by submitting PixelScript commands to the system. You can see the PixelScript commands corresponding to each menu item by looking at the files with a .tcl extension in the directory structure under `resources/menus`.

The Scripts Menu

You can place your scripts anywhere in the menu structure, however, there is a menu (the **Scripts** menu) specifically for custom scripts. PiXELS3D 4 includes several scripts in this menu to perform miscellaneous functions and to help show you how to write your own.

Reset

Resets the selected object's transformations or display color to their default values.

Reset Transformation. Resets the selected object's location, center, and rotation to 0,0,0 and sets the object's scale to 1, 1, 1.

Reset Position. Resets the selected object's position to 0, 0, 0.

Reset Rotation. Resets the selected object's rotation to 0, 0, 0.

Reset Scale. Resets the selected object's scale to 1, 1, 1.

Reset Color. Resets the selected object's display color to the default.

Fill Colors

Sets the display color of the selected object. This helps you differentiate objects when you are editing a scene.

Toggle Black

Switches between the default grey background with a grid in the camera view and a solid black background.

Pencil Test

Plays a quick preview of your animation so you can see how things move without doing a full rendering. The keyboard equivalent is `Cmd-k`.

The Scripts Menu

play.png lets you change the appearance of the controls in the timeline.

Figure 27.3: play.png



glic.png controls the appearance of the tool icons on the views.

Figure 27.4: glic.png

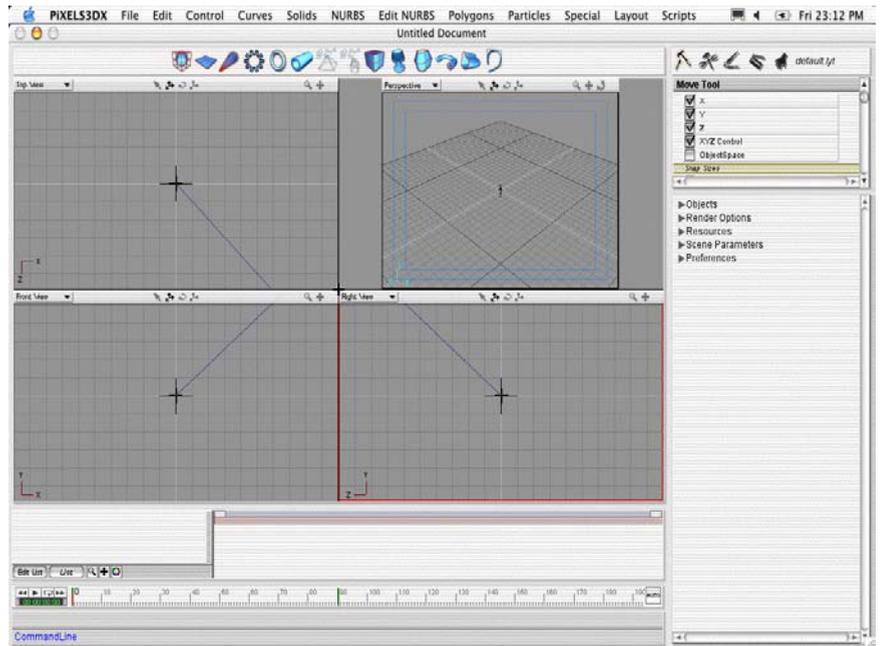


Note: glic.png is a 32 bit png file. 24 bits specify the color and 8 bits specify the alpha channel. Some png viewers or editors treat the extra 8 bits as colors. If you open the file and it looks mostly black, try a different viewer or editor.

For some of the interface themes there are corresponding scripts or menus. These alter the color of the view and on-screen text so they are clearly visible with the new theme. Copy scripts to the `resources:scripts` folder. Copy menus to the appropriate folder in `resources:menus` (for instance, since `New`

is on the File menu, new.tcl gets copied to resources:menus:0 | File:0 | New/N*)

Figure 27.5: Ice Theme



Appendix B. Legal Notice

PiXELS 3D Version 4 and previous versions are copyrighted 1996-2002 by Pixels Digital. All Rights Reserved.

U.S. Government End Users

PiXELS3D Version 4 (the Software) is a commercial item as defined in 48 C.F.R.01 (Oct. 1995), consisting of commercial computer software and commercial computer software documentation as used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government end users acquire the Software with only those rights set forth herein.

PiXELS3D Version 4 End User License Agreement (EULA)

By installing the software, you are consenting to be bound by this agreement. If you do not agree to all of the terms of this agreement, do not use the software. You may not redistribute this EULA.

Grant: PiXELS3D 4 grants you a non-exclusive license to use this version of the PiXELS3D software (the Software).

Warranty: The Software is provided on an AS IS basis, without warranty of any kind, including without limitation the warranties of merchantability, fitness for a particular purpose and non-infringement. The entire risk as to the quality and performance of the Software is borne by you. Should the Software prove defective, you and not PiXELS3D 4 assume the entire cost of any service and repair. In addition, the security mechanisms implemented by PiXELS3D software have inherent limitations, and you must determine that the Software sufficiently meets your requirements. This disclaimer of warranty constitutes an essential part of the agreement. SOME JURISDICTIONS

DO NOT ALLOW EXCLUSIONS OF AN IMPLIED WARRANTY, SO THIS DISCLAIMER MAY NOT APPLY TO YOU AND YOU MAY HAVE OTHER LEGAL RIGHTS THAT VARY BY JURISDICTION.

Limitations: You may not:

permit other individuals to use the Software except under the terms listed above,

permit concurrent use of the Software,

modify, translate, reverse engineer, decompile, disassemble (except to the extent applicable laws specifically prohibit such restriction), or create derivative works based on the Software,

copy the Software other than as specified above; rent, lease, grant a security interest in, or otherwise transfer rights to the Software, or

remove any proprietary notices or labels on the Software.

Title: Title, ownership rights, and intellectual property rights in the Software shall remain in PIXELS3D 4 and/or its suppliers. The Software is protected by the copyright laws of the United States and international copyright treaties. Title and related rights in the content accessed through the Software is the property of the applicable content owner and may be protected by applicable law. This license gives you no rights to such content. If you receive your first copy of the Software electronically and a second copy on media, the second copy may be used only for backup and archival purposes. This license does not grant you any right to any enhancement or update to the Software.

Termination: The license will terminate automatically if you fail to comply with the limitations described herein. Upon termination of this License, or in any event within

ten (10) days following PiXELS3D 4 release of a commercial version of the Software, you agree to destroy all copies of the Software.

Export Controls: You may not download or otherwise export or re-export the Software or any underlying information or technology except in full compliance with all United States and other applicable laws and regulations. None of the Software or underlying information or technology may be downloaded or otherwise exported or re-exported (i) into (or to a national or resident of) Cuba, Iraq, Libya, Yugoslavia, North Korea, Iran, Syria or any other country to which the U.S. has embargoed goods; or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nationals or the U.S. Commerce Department's Table of Denial Orders. By downloading or using the Software, you are agreeing to the foregoing and you are representing and warranting that you are not located in, under the control of, or a national or resident of any such country or on any such list.

Limitation of liability: Under no circumstances and under no legal theory, tort, contract, or otherwise, shall PiXELS3D 4 or its suppliers or resellers be liable to you or any other person for any indirect, special, incidental, or consequential damages of any character including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses. In no event will PiXELS3D 4 be liable for any damages, even if PiXELS3D 4 shall have been informed of the possibility of such damages, or for any claim by any other party. This limitation of liability shall not apply to liability for death or personal injury to the extent applicable law prohibits such limitation. Furthermore, some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this limitation and exclusion may not apply to you.

High Risk Activities: The Software is not fault-tolerant and is not designed, manufactured or intended for use or resale as online control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the Software could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). PiXELS3D 4 and its suppliers specifically disclaim any express or implied warranty of fitness for High Risk Activities.

Miscellaneous: This Agreement represents the complete agreement concerning this license and may be amended only by a written agreement executed by both parties. This license is personal to you and you agree not to assign your rights in herein, and any attempted assignment by you shall be null and void. If any provision of this Agreement is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This Agreement shall be governed by California law (except for conflict of law provisions). The United Nations Convention of Contracts for the International Sale of Goods is expressly excluded.

Index

Symbols

+ 26

Numerics

4 Pane View (on View menu) 16, 17

A

Accelerator 312

Active (checkbox for Force) 312

Affect Velocity (for Vortex Force) 315

Ambient Occlusion 186

Animation Layout 30

Animation menu 285

animation, creating 247

Aniso 166

arc

 creating 56

Atmosphere 187

atmosphere 167

Attach fCurve 285

Attractor 94

Auto 284

Auto Render (in Tempest) 266

auto-keyframes 284

B

Back View (on View menu) 17

Birth Rate (Editor) 308

Birth Rate Multiplier (Render) 308

blend 333

Blinn 166

Blobby 307

BMP, bitmap file 244

bones 322

Bottom View (on View menu) 16

Box (Force Shape) 312

Brick 191

Bucket Size 268

Build (Lattice) 321

Bulge 95

Bump 171

C

Cache images (in Tempest) 266

Camera View 15

Camera View (on View menu) 16

Camera View. 13

Camera, Selecting 17

circle

 creating 57

Clay 198

clay 178

Close Ends (on Edit NURBS menu) 100

Clouds 199

Collision 313

Color 170

color fields 181

Color Math 200

Color multiplication 174

cone

 creating 46

Constraints 288

control vertices 39

Cube 159

cube

- creating 53
- Cubic 103
- curves
 - arc 56
 - circle 57
 - gear 62
 - helix 59
 - ngon 61
 - polygon 61
 - square 58
 - star 60
- Cylinder 159
- cylinder
 - creating 44

D

- Death 307, 315
- Debug (in Tempest) 266
- decal 249
- default layout 13 14
- Default Screen Layout 13
- degree 103
- Delete (in ShaderManager) 244
- Delete (in TextureManager) 244
- Delete Texture Map (on Image Source menu in Image Map node) 244
- Dice Binary 268
- Diffuse 202
- diffuse 178
- Disk 158
- Displacement 171, 271
- Displacement Bounds 172, 272
- displacement seams 172
- Dynamics 308

E

- Edges Mode (on View menu) 17
- Edit NURBS, Close Ends 100
- Edit NURBS, Insert 96
- Edit NURBS, Invert 100
- Edit NURBS, Join 99
- Edit NURBS, Remove 97
- Edit NURBS, Reorder 100
- Edit NURBS, Shift Origin 100
- Edit NURBS, Simplify 98
- Edit NURBS, Subdivide 98
- Edit NURBS, Transpose 100
- Edit NURBS, Wrap 101
- Edit NURBS, Wrap and Project 102
- Emission Object 307
- Emission Radius 308
- Emit From 307
- emitter 303
- Expand 15
- Expand to Full View (on View menu) 15, 17
- Export (on File menu) 242, 243, 245
- Export to Movie (on File menu in Flipbook) 248
- Eye Splits 268

F

- Faces Mode (on View menu) 17
- faces, humanoid 119
- Favorites Bar 27, 42
- fBm 183, 204
- fCurve 286
- File Content (under Image Setup in Attribute-Manager) 246
- File Format (under Image Setup in AttributeM-anager) 246
- File menu 241

- File menu, Render 265
- F-key 26
- Flat Shaded 23
- Flipbook 245
- Floor 311
- fog 167
- Force 311
- Forces 309
- Four Pane View 17
- fps 283
- Fractile Sum 183, 206
- frame 283
- Frame Object 26
- framerate 283
- Friction 312
- Front View 13
- Front View (on View menu) 16
- Full View 17
- function keys 26

G

- gear
 - creating 62
- Global Coordinate 182
- glossy 167
- Granite 211
- Gravity 309, 313
- Gravity Direction 309
- Gravity Strength 310
- Gravity Well 313
- Grid Size 268
- grid size 20

H

- Hair 212

- Height (under Image Setup in AttributeManager) 245
- Height Map 270
- helix
 - creating 59
- Hide Selected (on View menu) 18
- Hide Unselected (on View menu) 18
- Hot Keys 21

I

- IK 322
- Image Cache (in Tempest) 266
- Image Map 212, 250, 271
- image map 169
- Image Setup (in AttributeManager) 245
- Import (on File menu) 242
- Insert (on Edit NURBS menu) 96
- installation 9
- Instance 306
- Inverse Kinematic 322
- Invert (for Collision Force) 314
- Invert (on Edit NURBS menu) 100

J

- Join (on Edit NURBS menu) 99
- JPEG file format 244

K

- Ka 182
- Kd 182
- keyboard shortcuts 21
- keyframe 283
- Knots 104
- Ks 182

L

- lattice 320
- Layout Manager 31
- Layout Selector 27
- layouts
 - default 13 14
- Left View (on View menu) 16
- Life 307, 316
- Lifespan 308
- Light, View From 17
- Limits 291
- Linear 103
- Load (in ShaderManager) 244
- Load (in TextureManager) 244
- Load Texture Map (on Image Source menu in Image Map node) 244
- Local Dynamics 308
- lofted blend 333
- Look At 290

M

- Marble 215
- materials 165
- Math 216
- Max Particles 307
- mesh
 - creating 49
- Mnplz CPU 266
- Modeling Layout 28
- Monopolize CPU 266
- morph targets 331
- mov file format 248
- Move Tool 19 20
- Movement Handles 19
- movie file, creating 247

N

- Name (under Image Setup in AttributeManager) 245
- Negative 273
- New (in ShaderManager) 243
- New (on File menu) 241
- New Lattice (on Special menu) 320
- New Morph Target 332
- New Particle System (on Particles menu) 303
- ngon
 - creating 61
- node, connect 173, 176
- node, new 173
- noise functions 183
- Noise Scale (for Turbulence Force) 315
- Numeric fields 181
- NURBS
 - cone 46
 - cube 53
 - cylinder 44
 - mesh 49
 - sphere 43
 - torus 51
- NURBS menu 42
- NURBS primitives 42

O

- Object Node 170
- object node 243
- Object Space (for Move Tool) 20
- Objects Mode (on View menu) 17
- Observe Object Home 26
- Octahedron 160
- Only Show Motion 303
- Only User Forces 309
- Opacity 171

Open (on File menu) 241
Open a Sequence (on File menu in Flipbook)
247
Orbit 291
orthogonal projection 14

P

P 183
Pan 21
Park 320
particle system 303
Particle Type 304
Particles menu, New Particle System 303
Particles Travel 308
Patch, Reorder 100
Patch, Simplify 98
pct file format 244
periodic 40
Perspective From Light (on View menu) 17
perspective projection 15
Phong 165
PICT file format 244
Pixel Aspect Ratio (under Image Setup in At-
tributeManager) 245
Pixel Samples 267
PixelScript 339
Play Sounds (in Tempest) 266
PNG file format 244
Points 304
polygon
 creating 61
Posterize 220
psd, Photoshop document file 244

Q

Quadratic 103

Queue.dat 266
QuickTime image format 244
QuickTime movie 248
QuickTime Pro 247
Quintic 103

R

ray marching 167
registration 9
Remove (on Edit NURBS menu) 97
Render (on File menu) 243, 265
Render Animation (under Animation in Attrib-
uteManager) 246
Render Options 172, 267
Render Options (in AttributeManager) 245
Renderman 242
Reorder (on Edit NURBS menu) 100
rgb 246
rgba 246
rgbaz 246
rgbz 246
RIB 242
Right View 13
Right View (on View menu) 16
Rotate Tool 20
Round, Close Ends 100

S

s 184, 223
Save (in ShaderMaker) 244
Save (on File menu) 241
Save As (in ShaderMaker) 244
Save As (on File menu) 241
Scene Dynamics 309
Screen Square 304
Scripts menu 340

- seams 172
- Select 104
- Select Camera (on View menu) 17
- Select Mode (for Select Tool) 18
- Select Tool 18
- Select U Columns 25
- Select V Columns 26
- Selection fields 181
- Semi-Uniform 104
- Set Reference Pose 332
- SGI image file format 246
- Shaded Lit 24
- Shaded Unlit 24
- shader 165, 243
- Shader Layout 243
- shader, load 168
- shader, new 168
- ShaderMaker 169
- ShaderMaker Pro 243
- ShaderManager 167, 243
- Shading Layout 29
- Shading Rate 267
- Shape (for Force) 312
- Sharp, Close Ends 100
- Shift Origin (on Edit NURBS menu) 100
- Show All (on View menu) 18
- Simplify (on Edit NURBS menu) 98
- Skin 224
- skin 178
- SM3 Surface 224
- sm3_surface 178
- smp, ShaderMaker Pro format 243
- Snap 291
- Snap To Grid (for Move Tool) 20
- specular function 165
- specular highlight 165
- Sphere 158, 306
- sphere
 - creating 43
- Spherical (Force Shape) 312
- spline deformer 319
- Spline, Reorder 100
- Spots 225
- Square 157
- square
 - creating 58
- star
 - creating 60
- steps 41
- Stipple Surfaces 25
- stripes, creating 230
- Subdivide (on Edit NURBS menu) 98
- Subsurface Scatter 167
- Surface 225
- surface 178
- Surface Nodes 178
- system requirements 7

T

- t 184, 226
- Tag Map 245, 249
- Targa file format 244
- Tcl 339
- Tempest 172, 265
- Tetrahedron 305
- Texture 249
- Textured 24
- TextureManager 168
- tga file format 244
- TIFF file format 244
- Tile 227
- Timeline 283
- Top View 13
- Top View (on View menu) 16

torus
 creating 51
Transpose (on Edit NURBS menu) 100
Triangle 157
Turbulence 183, 228, 315
tweens 284
Two Sided 268

U

U Back, Shift Origin 100
U Columns (for Select Tool) 18
u dimension 40
U Forward, Shift Origin 100
u row 41
u step, insert 96
u Step, Remove 97
U to U, Join 99
U, Invert 100
U, Simplify 98
Uniform 104
User Defined 229

V

V Back, Shift Origin 100
v column 41
V Columns (for Select Tool) 18
v dimension 40
V Forward, Shift Origin 100
v step, insert 96
v Step, Remove 97
V to V, Join 99
V, Invert 100
V, Simplify 98
Vector fields 181
Velocity 308

Velvet 230
velvet 178
verbose (in Tempest) 266
Vertices Mode (on View menu) 17
View menu 16
View menu, 4 Pane View 16
View menu, Expand to Full View 15
View Pane context menu 22
View Panes 13, 15
view panes 14 26
 controls 15
 sizing 15
volumetric lighting 189
Volumetrics 167
Vortex 314

W

Wave2D 230
Wavefront, OBJ file 241
Weave 232
Widget 30
Width (under Image Setup in AttributeManager) 245
Wind 310, 316
Wind Direction 310
Wind Strength 311
Wireframe 23
Wood 234
Wrap (on Edit NURBS menu) 101
Wrap and Project (on Edit NURBS menu) 102, 335

X

X (for Move Tool) 20
XYZ Control (for Move Tool) 20

Y

Y (for Move Tool) 20

Z

Z (for Move Tool) 20

z buffer 246

Zero Gestures 332

Zoom 21

Zoom Tool 21