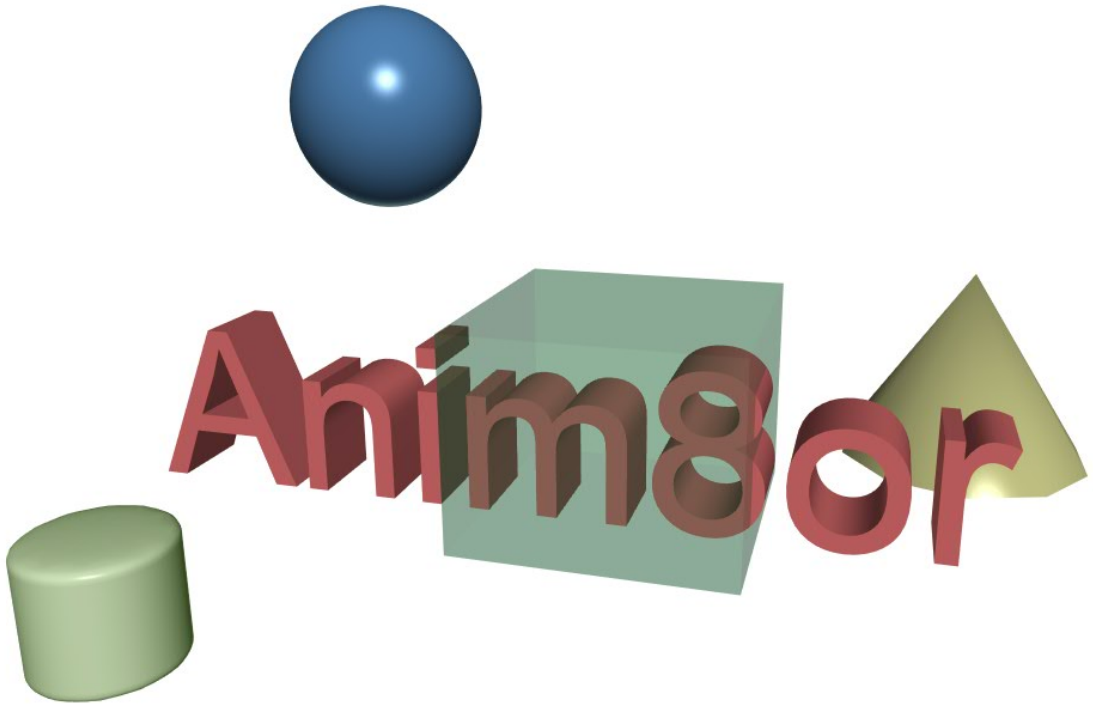
Anim8or®

User Guide



Version 1.00 29-May-17

R. Steven Glanville

© 1999 - 2017 R. Steven Glanville. All rights reserved. http://www.anim8or.com

Table Of Contents

Introduction **9**

What's New in v1.0 10

User Interface 10

Modeling 10

Animation 11

Basics **12**

Mouse Usage 12

Keyboard Shortcuts 12

Undo and Redo 12

Tool Tips 13

Toolbars and Menus 13

Common Button Meanings 13

Top Toolbar 15

Arc Rotate 16

Editing Widgets 16

Grid Control 17

Material Editor 17

Anim8or Object Libraries 18

Visual Quality 18

OpenGL Workspace 19

File Output 20

User Attributes 20

Printing 20

Auto Save 20

Configuration 21

 © 1999 - 2017 R. Steven Glanville

User Interface Configuration 22

Object Editor - Basics & Object/Edit Mode 24

Object/Edit Tools 25

Basic Objects 25

Mesh vs. Parametric Components 27

Object Materials 28

Plug-in Shapes 28

Splines 29

True Type Fonts 29

Extrusion 30

Lathing 31

Modifiers 32

Mirror Image 33

Morph Targets 34

Continuously Mirrored Meshes 35

Reference Image 35

Object Editor - Object/Point Mode 37

Object/Point Operations 37

Point Editing 37

Edge Editing 38

Face Editing 39

Applying Multiple Materials 39

A Note on Selecting Faces 40

Adding Points and Edges 40

Adding Faces 41

Connecting Meshes 41

Merging Points 42

 © 1999 - 2017 R. Steven Glanville

Connecting Meshes (2) 42
Connecting Meshes (3) 42
Point and Line Parameters 43
Face Extrusion and Manipulation Tools 43
Topographical Knife 46
Face Editing Commands 47
Miscellaneous Commands 49
Selection Commands 49
Editing Commands 49
Figure Editor 51
Figure Basics 52
Figure/Edit Operations 52
Visibility 54
Building a Skeleton 55
Flexible Joints 56
Adding Body Parts 57
Skinning 57
Influence Volumes 58
Skinning Weights 59
Sequence Editor 61
Time Track 61
Scrubber Bar 62
Sequence Basics 62
Edit Operations 63
Visibility Options 63
Animate Button 64
What is a Key? 64

 © 1999 - 2017 R. Steven Glanville

Making a Key Pose 64
Using the Trackball 65
Click-dragging on Bones 65
Inverse Kinematics 65
Editing Key Frames 66
A Circle Has 720 Degrees 66
Ghost Views 67
Making a Sequence 67
Scene Editor 68
Elements of a Scene 69
Adding Objects 69
Adding Figures 70
The Camera 70
Lights 70
Targets 71
Object Properties Dialog 71
Light Properties Dialog 73
Advanced Light Properties Dialog 74
Shadow Properties of Lights 74
Shadows 74
Volume Shadows 75
Ray Traced Shadows 75
Environment Settings 76
The Time Track 78
Previewing and Saving Images 79
The Alpha Channel 79
Image Compositing 80

 © 1999 - 2017 R. Steven Glanville

Rendering an .AVI Movie 81
Alternate Renderers 81
Animation **82**
Animation with Key Frames 82
Animating with Inverse Kinematics 82
Inverse Kinematic Walk Cycle 83
Animation Figures with Sequences 84
Animating with Expressions 84
Controllers 85
Kinds of Controllers 85
Editing Keys 85
Editing Expressions 86
Graph Editor 87
Editing Segments 87
Materials **89**
Material Editor 89
Texture Selector 93
Advanced Textures 94
Normal Maps 95
Environment Maps 96
Texture Mode 98
Scripts **100**
Creating a Script 100
The Script Directory 100
Installing a Script 101
Plug-in Scripts 101
Parametric Plug-in Scripts 102

 © 1999 - 2017 R. Steven Glanville

Export Plug-in Scripts 102
Running a Modeling Script 102
Controller Scripts 102
Script Errors 103
Writing Scripts 103
Data Types 103
Variables 104
Expressions 104
Function Calls 104
Unary Operators 104
Binary Operators 105
Statements 105
User Functions 106
User Attributes 106
ART Ray Tracer 108
Art Materials 108
Class anim8or 108
Class glossyreflector 109
Class transparent 110
Class dielectric 110
Light Attributes 111
Ambient Occlusion 112
Ray Tracing and Anti-Aliasing 112
Keyboard Shortcuts 114
Menu Shortcuts 114
Viewpoint Shortcuts 114
Top Toolbar Shortcuts 114

 © 1999 - 2017 R. Steven Glanville

Object Editor Shortcuts 115

Point Editor Shortcuts 115

Figure Editor Shortcuts 116

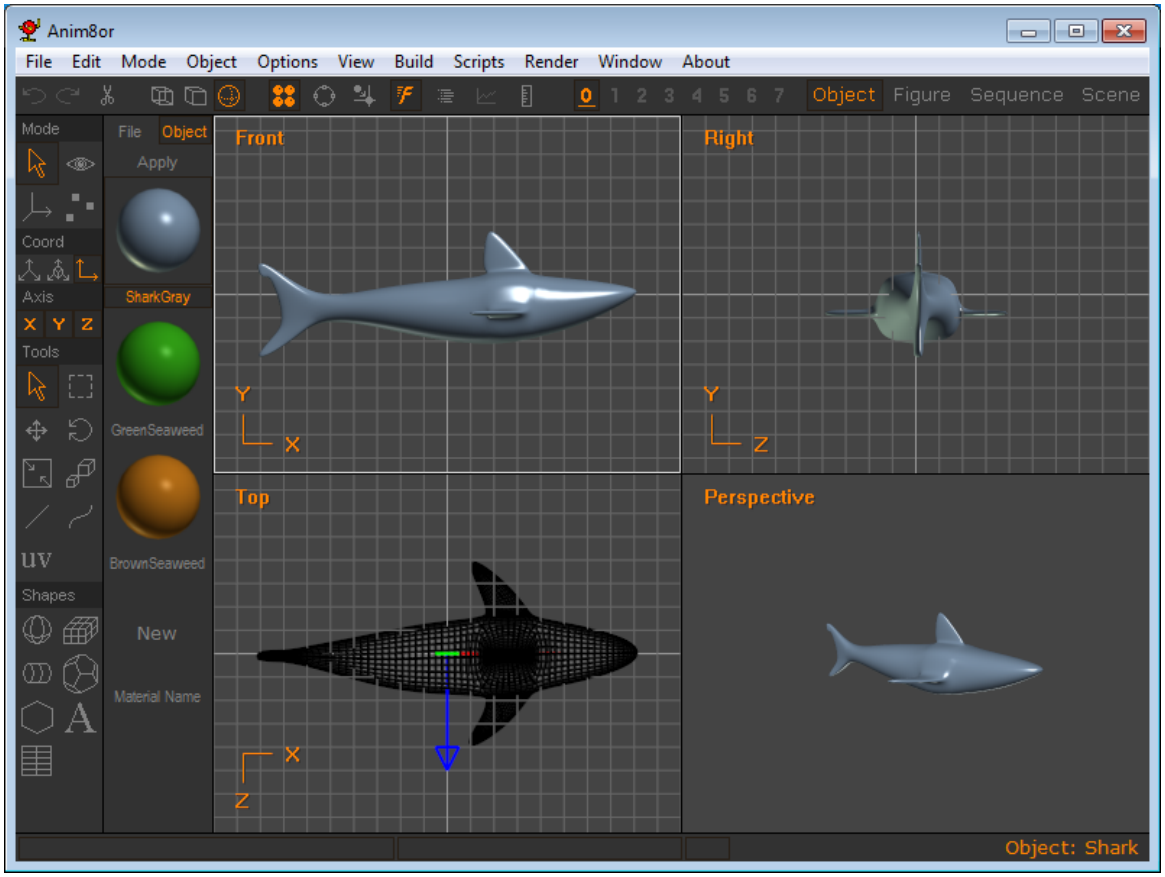
Sequence Editor Shortcuts 117

Scene Editor Shortcuts 117

 © 1999 - 2017 R. Steven Glanville

Introduction

Anim8or is a 3D computer animation program that is designed to allow straightforward creation of animations. You interactively create and edit objects, figures, and scenes directly on the computer's screen. The basic interface is similar to most 3D animation and CAD programs, as shown in the following screen shot:



You control various aspects of your work using an ordinary computer mouse or tablet. You select and drag, rotate, scale and place objects by clicking in the various views of your work. There are two **toolbars** that you can use for common tasks. The one at the top of the screen has buttons for general commands that are used throughout Anim8or, while the one on the left allows you to change your mode of operation for common tasks in the current editor that you are using. The usual **menu** is there for less frequently used tasks.

You can control the view of your objects to see how they appear from the **front**, **left**, **top**, **back**, **perspective**, etc., and you can show either multiple views or a single view on your screen.

There are four editors.

First, there is an **object editor** that is used to build (mostly) static objects. This is where you create and refine the basic models for your 3D world.

Second, there is a **figure** or **character editor**. You use it to define the structure of a character that you want to animate by connecting jointed **bones** together, and attaching objects to them. You can give each joint different ways in which they are allowed to move and restrict the range, just like your elbow or shoulder can move only in certain ways.

© 1999 - 2017 R. Steven Glanville

Third, there is a movement or **sequence editor**. Here you define segments of motion, such as one cycle of a walk. These can then be linked together in chains in the scene editor for longer sequences.

Finally, you put together your final scenes in a **scene editor**. Here you place the objects and built in the other parts of Anim8or into their final "world". You can control how they move, and where the **camera** is located. You can also give the scene various kinds of **lights**.

Once you have created a model or scene you can render high quality **.jpg**, **.bmp** and **.png images** and **.avi movies** and save them to disk for later use.

Finally, you can use Anim8or for 3D printed objects. You can import and export fines in the **.stl** format used by 3d printers.

What's New in v1.0

There are several major additions to Anim8or in version 1.0 for both the modeling and animation. They are summarized here and described in more detail in their respective sections throughout the manual.

User Interface

The user interface has several enhancements that can streamline modeling and animation. The most obvious is the new color scheme which has a **dark gray background** instead of light blue. This makes models' shapes generally easier to see. You can also increase the size of the tool-bar buttons which helps touch-screen users. Additionally, with an adequate graphics chip, you can change the 3D views to **anti-aliased**, and show edges and controls in **thicker lines**, which can make them easier to see on high resolution screens.

Enabling **fast-select** allows a more fluid selection and editing work flow since it eliminates the need to continually switch modes to edit different parts of a mesh or scene. Moving the mouse highlights in orange the component that would be affected if you click making it easier to select the correct point or edge.

Layers are an addition that lets you assign components to different groups that can be locked or hidden by the click of a button.

You can temporarily enter **arc-rotate** mode by pressing and holding the Alt key, and you can **zoom** the viewport by scrolling the mouse wheel.

Modeling

There are several improvements to the modeler. The most important is the **topographic tool** or **topo-knife**. You can split faces, and move, add and dissolve points and edges all in one tool.

Scaling and rotating faces, edges and points now does so relative to the center of the group that's selected, rather than the center axis of the mesh.

© 1999 - 2017 R. Steven Glanville

Animation

The main addition to Anim8or's animation capabilities is **Inverse Kinematics**, or **IK**. This allows you to position a chain of multiple bones at the same time. You simply position the end of the one bone where you want it and all of the other bone in the chain adjust. You can also lock a foot to the ground, move the body and the leg will adjust as needed.

© 1999 - 2017 R. Steven Glanville

Chapter 2 - Basics

Basics

Mouse Usage

Anim8or requires the use of a mouse with two buttons, but using one with a middle button can simplify some commands. You use the left one most of the time to select, move, and change objects. When selecting, the **left button** always deselects everything beforehand, so that only a **single item** remains selected. The **right button**, on the other hand, allows you to select **multiple items** by adding newly selected ones to those already selected. You can use the **middle button** to **deselect** specific objects. If you don't have a middle button some operations will require a few more steps.

When moving, rotating, or scaling something non-uniformly, the **left button** changes the **X**-axis and **Y**-axis. Moving to the left or right affect the X-axis, and up and down the Y. The **right button** changes the **Z**-axis.

You can usually **double click** on something to bring up a dialog to view and edit edit its properties.

Keyboard Shortcuts

Many tools and commands are associated with **keyboard shortcuts** or **hot keys**. These are shown in this manual enclosed by square brackets [***L***] next to the button or command. You can press this key instead of clicking on a button or selecting a command from the menu. They are summarized in *Appendix A: Keyboard Shortcuts*.

Undo and Redo

Anim8or has a complete **undo** command for the Object, Figure, Sequence and Scene Editors. As long as you stay in either Editor you can undo and **redo** multiple editing commands. A history is kept of as many changes as can fit in the undo buffer, initially set to about 25 MB of memory. You can increase this in the Configuration dialog. So unless you're editing large models, you will be able to undo more than one operation.

You use the Edit→Undo and Edit→Redo menu commands to apply undo's and redo's. Or you can use the top toolbar buttons  [***Ctrl-Z***] and  [***Ctrl-Y***].

If you wish to cancel a currently active operation, such as if you are dragging the mouse to change an Object's orientation, you can still use the original (rather simplistic) substitute. When you are dragging a mouse in the main window to change something, say an object's orientation, and you think "Oops! I didn't want to do that!" just keep on dragging the mouse until it's outside the active view window and release the button. This will cancel the operation.

 © 1999 - 2017 R. Steven Glanville

Still, when using this and any other computer program, I still offer the same advice: "Save Early, Save Often!" And don't always save to the same file. Keep a recent backup handy.

Tool Tips

You can turn tool tips on with the Options→ToolTips menu command. Then if you pause the mouse cursor over a toolbar button it function will be shown in the status bar.

Toolbars and Menus

Here is a portion of a typical toolbar. It has several sections in it that control different aspects of its behavior. The first thing that you will notice is that markings on some of the buttons are in **ORANGE**  , some are in **GRAY**  , and a few are in **DARK GRAY**  . The orange ones indicate that the associated mode or option is currently selected or in effect. The gray ones indicate a state that is not selected. The dark gray buttons show options that are disabled.

The top group of buttons on each toolbar act as a radio set indicating the current editing mode of the active view. When you click on any one of them, it becomes highlighted and that mode is selected. All of the others are deselected. The meanings of the various buttons in this group, and the other common buttons, are described below.

Common Button Meanings

The topmost group can have up to 4 buttons in it. When you click on one it sets the indicated mode, and changes the lower part of the toolbar to show the functions that are present in the new mode. The different modes are:

[***A***] This button indicates that you are in the most basic select-and-edit mode for the editor that you are currently using. You will use it for most operations, such as selecting things and common editing actions.

[***V***] This button is used for changing your view or views of your workspace. With it you can pan, zoom, tilt, and scale each view independently.

[***O***] In this mode you can move and rotate the pivot for objects in your workspace. A pivot is the point and orientation that are used when you scale and rotate each object.

 © 1999 - 2017 R. Steven Glanville

[***P***] You use this button to move into point edit mode. It allows you to add, move, and modify individual points, edges, and faces of editable mesh objects.

The next row of three small buttons 

shows the three axes of the 3D world. They indicate which axes are "unlocked", allowing you to move and rotate things in each direction. You can individually select or deselect them, and thus restrict the movement of objects that you are changing.

Below these buttons you will find three more small buttons  of which only one can be active at a time. They show the current coordinate system you are using. You can use **world**, **object**, or **screen** coordinates when you manipulate things.

You will find that there are several common toolbar buttons that have the same or similar meanings in more than one editing mode. They are described next.

Select [***a***] - You use the arrow button to select individual components. In this mode clicking on an object with the left mouse button selects it and deselects previously selected objects. Right-clicking selects additional objects without deselecting currently selected ones.

Tip: holding down the **Ctrl** key then pressing the mouse button allows you to select multiple objects using the mouse as a sort of paintbrush.

Drag-Select [***d***] - With this button you can click and drag to show the corners of a selection rectangle. Any and all objects that fall within that area are then selected. Again, the left mouse deselects any previously selection items, while the right mouse does not.

Move [***m***] - This is the move function. You set it when you want to move your objects around on the screen. When you press the left mouse button within a view window you can drag anything that is currently selected in the X and Y directions by moving the mouse to the left and right, or up and down. The objects will move along with your mouse. Similarly, the right mouse button will move them along the Z-axis, which is usually into the screen.

Rotate [***r***] - This is the rotate button, used for rotating things. When you click and drag the left mouse button, any selected objects will rotate around the X and Y-axes along with your mouse. You can use the right button to rotate them around the Z-axis.

Non-uniform Scale [***n***] - This is the non-uniform scaling button. With it you can scale most objects independently in the X any Y directions using the left mouse button, and in the Z direction with the right mouse button.

Scale [***s***] - This button scales objects in all dimensions simultaneously.

In the Sequence and Scene editors there is set of buttons for controlling the playback of your animations. You use its buttons in the same manner as those on a video player.

 © 1999 - 2017 R. Steven Glanville

Top Toolbar

This is a toolbar that's at the top of the window. You can use it for several tasks common throughout Anim8or.

The buttons on the top toolbar do the following things:

Undo [***Ctrl-Z***] - If this button is enabled you can use it to undo the most recent edit command.

Redo [***Ctrl-Y***] - Reapply a command that you have un-done.

Cut [***Ctrl-X***] - Deletes any selected components and places them on the clipboard.

Wireframe [***Ctrl-W***] - Changes the currently active view of your object to Wireframe.

Flat [***Ctrl-F***] - Changes the view to Flat or Faceted.

Smooth [***Ctrl-S***] - View objects as smooth.

Material [***Ctrl-M***] - Toggles the display of the Material toolbar.

Arc Rotate [***Ctrl-R***] - Enables the Arc Rotate tool. You can easily pan, zoom, and rotate any view with Arc Rotate. It is described in the next section.

Snap to Grid [***Ctrl-G***] - You can toggle Snap to Grid mode with this button.

Fast Select [***Ctrl-T***] - When this is enabled you can click on unselected objects and immedi-ately edit them. Experienced users often prefer this mode because it allows faster editing, while newer users find it easier to first select, and then edit things.

List Items [***Ctrl-L***] - This button will open a text window listing the objects in the current view. You can click on them individually by name to select or deselect them.

Graph Editor [***Ctrl-P***] - This button will open a window with a graphical view on the keys in your Scene or Sequence. You can use the mouse to edit keys here.

CAD Notations [***Ctrl-D***] - Displays additional measurement notations on the selected object.

Layers - There are 8 editing layers which can be locked or hidden. You can as-sign individual components to any one of the 8 layers. Clicking on the layer button toggles that layer's visibility. If the button for the layer is grayed out then it is hidden, and dark gray layers don't contain anything. Right-clicking locks and unlocks a layer, indicated by a small lock next to the number. Objects in locked layers are visible but you cannot select or edit them.

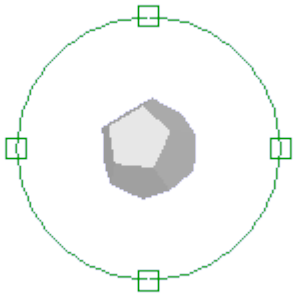
 © 1999 - 2017 R. Steven Glanville

The right portion of the top toolbar has four buttons that allow you to quickly switch between different editors:

You can also use **Shift+RightArrow** and **Shift+LeftArrow** to move between editors.

Arc Rotate

When you press and hold the **Alt** button Anim8or temporarily enters **arc rotate** mode and displays a special overlay over the focus window. You can then use the mouse to pan, scale and rotate your view. It may take you a try or two to get used to how it works, but once you have it can save you a lot of time over other methods. You can also toggle arc rotate by clicking on the arc rotate button  [***Ctrl-R***] in the top toolbar.

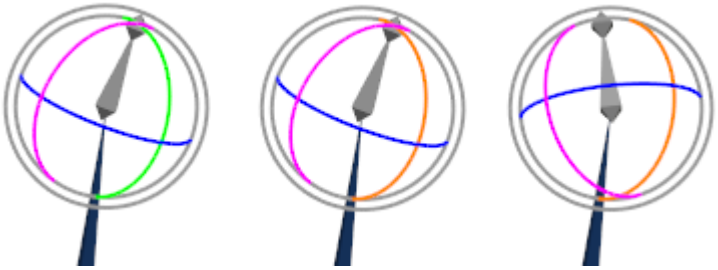


To the right is a view of the arc rotate screen. There are six different regions on the screen, separated by the green overlay: the inner, outer, top and bottom, and left and right. Each behaves differently when you click your mouse in them, and each mouse button controls a different function.

- Left button - Rotate the view. Clicking in the center rotates it around the X and Y-axes. The outer region rotates around the Z-axis. Clicking inside one of the four little squares rotates in the X-axis (for the top and bottom) or the Y-axis (for the left and right) only.
- Right button - Dolly the view. The center area moves your view left and right, and the outer region moves into and out of the screen. You may not see any change when dollying into the screen in orthogonal views but they will be apparent in a perspective view.
- Middle button - Scale the view. The different areas all behave the same when you are scaling a view.

Editing Widgets

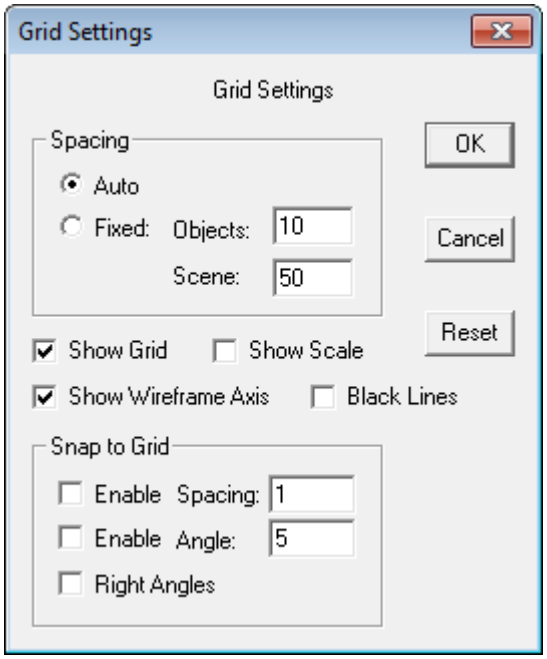
New to version 1.0 of Anim8or, **widgets** are on-screen controls that both simplify and offer better control of changes you make. Shown below is a trackball that is used to rotate bones around individual axes. The colored arcs show you which rotations are available. Green represents a rotation around the X-axis, blue the Y-axis, and magenta the Z-axis. Moving the mouse over the highlights the available controls.



 © 1999 - 2017 R. Steven Glanville

Grid Control

You can set the size of the background grid used in the editing views, or you can let Anim8or pick a default size that changes with how much you zoom the view so that it always shows a reasonable number of grid lines. Use the Options→Grid command to show the grid dialog.



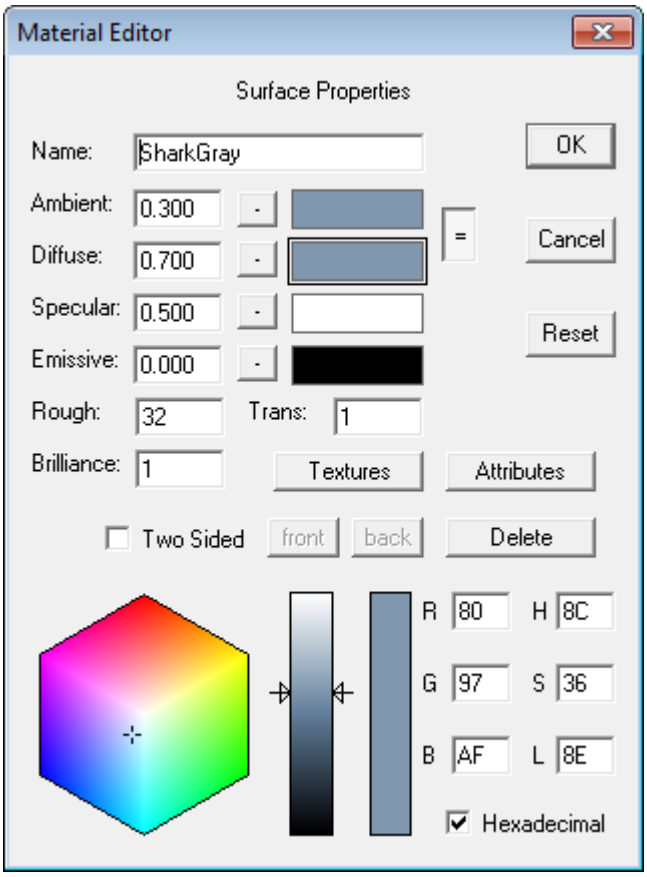
There are two grid sizes that you can set. One is used in the Object, Figure, and Sequence Editors, and the other is used in the Scene Editor.

You can also set snap-to-grid for the Object editor by checking the Enable box in the Snap to Grid section, or with the  snap to grid button [***Ctrl-G***] in the top toolbar. This will position objects you move at locations on a grid, making it easy to align them. Snap-to-grid also controls the size and rotation for many operations to give you more control on other aspects of modeling. You set the size of the grid snap in the Spacing entry. The Right Angles box is useful for adding edges at precise 90 degree angles.

Material Editor

The material editor is used whenever you need to design the color, texture, transparency, and other visual properties of an objects appearance. An example is shown below:

© 1999 - 2017 R. Steven Glanville



Materials are a complex topic. This dialog has are several sub-dialogs for certain functions. See *Chapter 9: Materials* for more details on how to use the Material dialogs.

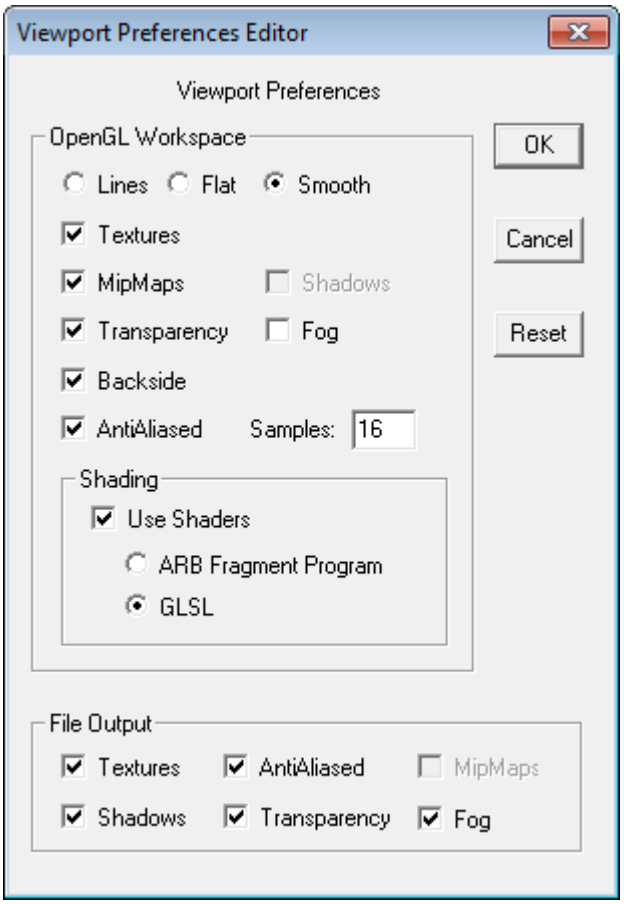
Anim8or Object Libraries

You can use any Anim8or **.an8** project file as a library. Objects, Figures, Sequences, and Scenes can all be imported to another Anim8or project with the Object→Import, Figure→Import, etc. menu commands. You can also export individual Objects and such with similar Object→Export commands. Make sure to select the **.an8** file type.

Visual Quality

You can control the quality of the images shown when you edit, as well as the quality of the images output to files. These are controlled through the Viewport Preferences dialog that is found under the View→Preferences menu command. It looks like:

 © 1999 - 2017 R. Steven Glanville



OpenGL Workspace

This section controls the appearance of your interactive workspace. You can view your objects as wireframe models, as flat faceted solid objects, or as smooth shaded solid objects.

Most graphics cards display smooth shaded images at the same speed as flat shaded ones, so you might think that flat shading wouldn't be very useful. But it is sometimes helpful to see individual facets on a model.

Texturing, and mipmaps, can sometimes slow the response of your computer, so use the check boxes to enable or disable them as you see fit. Also, texturing can hide the structure of your models, so you don't always want to see it.

You can add a general **fog** to Anim8or Scenes. Objects beyond a certain distance gradually fade into the color of the fog. Fog is only visible in the camera view of the Scene editor.

Most computers support **anti-aliased** 3D graphics directly in hardware. Checking this box changes the display to use that mode. You may need to restart Anim8or for to be effective.

 © 1999 - 2017 R. Steven Glanville

The **Use Shaders** check box enables drawing higher quality materials in the working views. It is available only on computers that support OpenGL pixel shaders. Graphics cards that support DX9 or higher will usually work. If your graphics card doesn't have this feature the check box will be disabled.

File Output

This section controls the appearance of your interactive workspace. High quality images are rendered using a software renderer or a ray tracer instead of an OpenGL accelerator, so the speed of your file renderings can be affected quite severely by the settings that you choose.

Anti-aliasing helps to remove jaggies but takes several times longer to do.

User Attributes

You can add your own unique numeric and non-numeric properties called **attributes** to your models, materials, characters, etc. Attributes are animatable. While they are primarily useful in scripts, they are quite flexible and are sometimes used to help develop new features for Anim8or. Attributes are saved in the .an8 output files so they can also be used to share data with auxiliary programs that read the .an8 format. See *Chapter 10 - Scripts* for more details on attributes.

Saving Images as .jpg, .png and .bmp Files

You can make a quick preview image of any view that you are working on with the File→Render-Preview command. You can also save this image to a **.jpg**, **.png**, or **.bmp** file, or print it, from within this command.

Printing

Anim8or has the usual Print and Print Preview options in the File menu. But it offers one additional level of control of your printed images, by letting you control the size and other properties of the bitmap image used in printing. It is not at all uncommon for a color printer to have a resolution of 1440 by 720 dpi. But generating an 8.5" by 11" image at this resolution is not practical! Instead Anim8or uses a default size of 1024x768 which you can adjust to suit your needs.

Auto Save

There is an Auto Save feature in Anim8or that will save a backup copy of your project at regular intervals when the project has been modified. You can set the interval, which defaults to 5 minutes, in the File→Configure dialog.

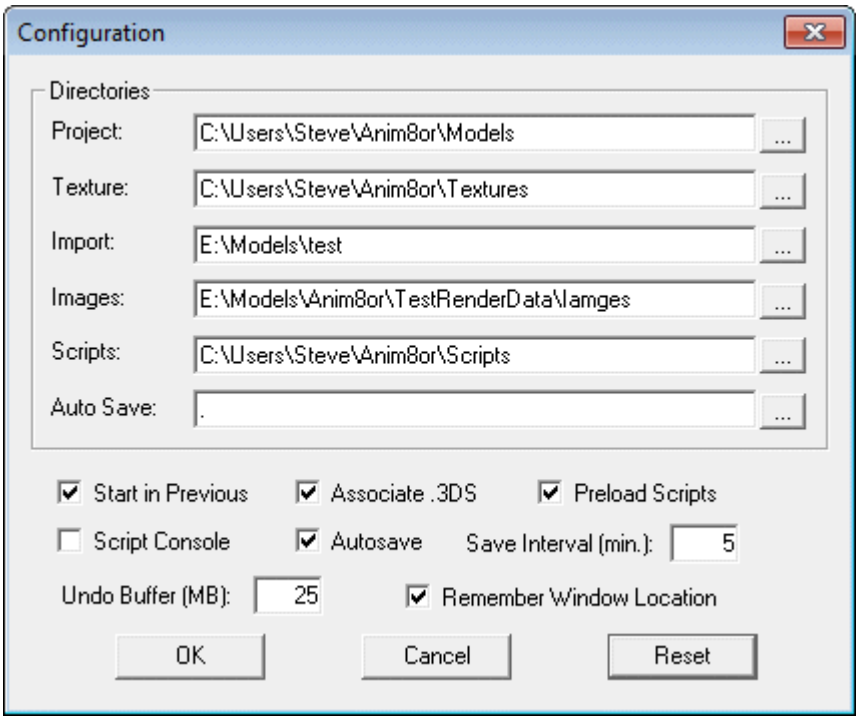
The name of the file is _autosave00000000.an8 or something similar. You can set the directory that the back-up file is written to as well. The default is the current working directory.

Anim8or does not automatically try to restore the file after a crash, or notify you that one exists. But remember to look for one when you've had a problem.

 © 1999 - 2017 R. Steven Glanville

Configuration

Anim8or keeps track of several base working directories. One is used to store Anim8or project files (.an8) and the others are the default directories for keeping textures, importing and exporting objects, saving pictures, and to store scripts which you can use to customize Anim8or in various ways. You set these directories with the Configuration dialog through the File→Configure command:



When you load or save a project file Anim8or uses the project directory as the initial default. You can change this directory in the Configuration dialog.

When you load a project file, Anim8or looks for texture files in three directories. First it looks in the same directory as the .an8 file. If that fails it tries the Texture directory set in the Configuration dialog. As a last resort Anim8or uses the full path name stored in the project file, which is where the texture file was initially found.

The import directory is simply the default place for importing and exporting files of any format.

The images directory if the default location for saving any images or movies that you render.

The scripts directory is where Anim8or looks for scripts to preload. This is where you keep plug-ins as well since they are scripts. Scripts are only preloaded if the Preload Scripts box is checked.

You can have Anim8or automatically load the last project you were using by checking the "Start in Previous" box.

You can enable Auto Save by checking the box labeled accordingly and set the time interval of the saves.

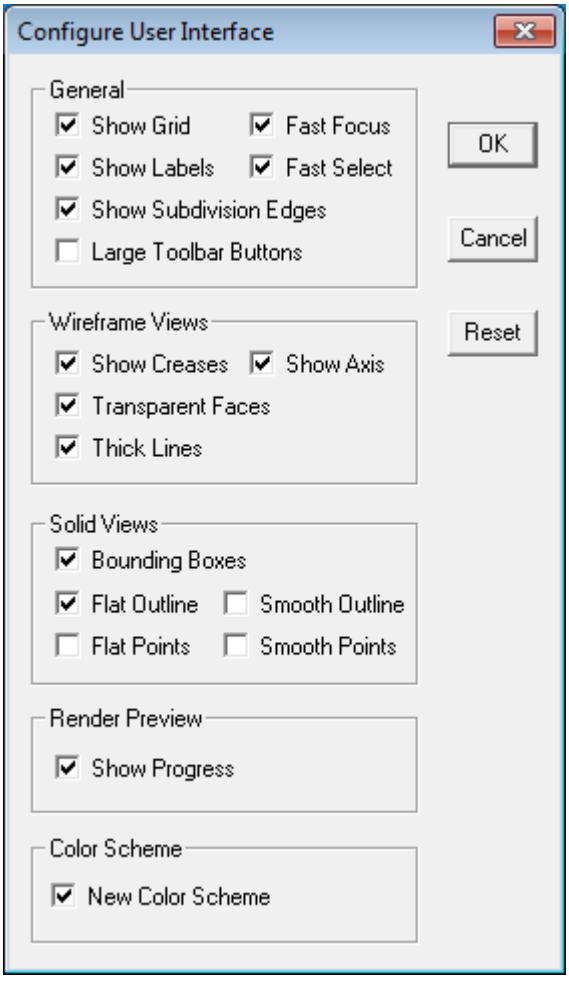
© 1999 - 2017 R. Steven Glanville

If you check the Associate .3DS box then you will be able to start Anim8or by double clicking on .3ds files in Windows.

User Interface Configuration

You can change certain aspects of the way that the Anim8or working views look with the User Interface Configuration dialog. You can find it under the File→Configure-UI command:

Most entries have obvious meanings but you might need an explanation for a few of them.



Fast Focus only applies when you have multiple working views visible at the same time. It controls what happens when you click inside a window other than the one with the mouse focus. If fast focus is enabled then the current active editing command will be applied to the new window on the first click. So if you are in Select mode then any object you click on in the new window will be immediately selected. If fast focus is disabled then the first click will only change the focus window. You will have to click a second time to select something.

Fast Select allows you to both select and edit components with a single mouse click. It can be somewhat difficult to use for newcomers to Anim8or. If you disable Fast Select you will generally need to change to one of the select tools to select components, and then back to the tool you want to use, but you are unlikely to accidentally deselect your work by mistake. You can also hold down the **Shift+Ctrl** keys to temporarily switch to Arrow-Select mode.

Large Tool Buttons increases the size of the buttons in the toolbars. Users that have a touch screen often find this helpful.

Transparent Faces in the Wireframe Views section controls the drawing of a thin, transparent skin on the faces of a model in wireframe viewing mode. This is on by default but you may sometimes find it easier to edit a particular mesh with it disabled.

You can show the wireframe **outline** or the vertex **points** overlaid on your solid models in both the flat and smooth viewing modes by checking the appropriate boxes in the Solid Views area. You may find that showing all this information can slow your computer down, especially if you are editing complex models, or make your model difficult to see. If this happens try disabling these features to improve performance.

© 1999 - 2017 R. Steven Glanville

Thick Lines. On very high resolution screens the lines used to show mesh edges and control structures can be difficult to see. Enabling Thick Lines changes makes them wider and easier to see. Enabling anti-aliased viewports as well can dramatically improve the appearance of the 3D workspace.

You can show the wireframe **outline** or the vertex **points** overlaid on your solid models in both the flat and smooth viewing modes by checking the appropriate boxes in the Solid Views area. You may find that showing all this in-formation can slow your computer down, especially if you are editing com-plex models, or make your model difficult to see. If this happens try disabling these features to improve performance.

When you render an image Anim8or normally updates the screen to show the progress of the render. Some graphics cards are very slow when updating the screen and this can increase the time it takes to render an image considerably. You can disable these partial image updates by unchecking the **Show Progress** box in the Render Preview section. Anim8or will still display a progress bar so you will still see how far the render has progressed. The final image will still be displayed when it is finished.

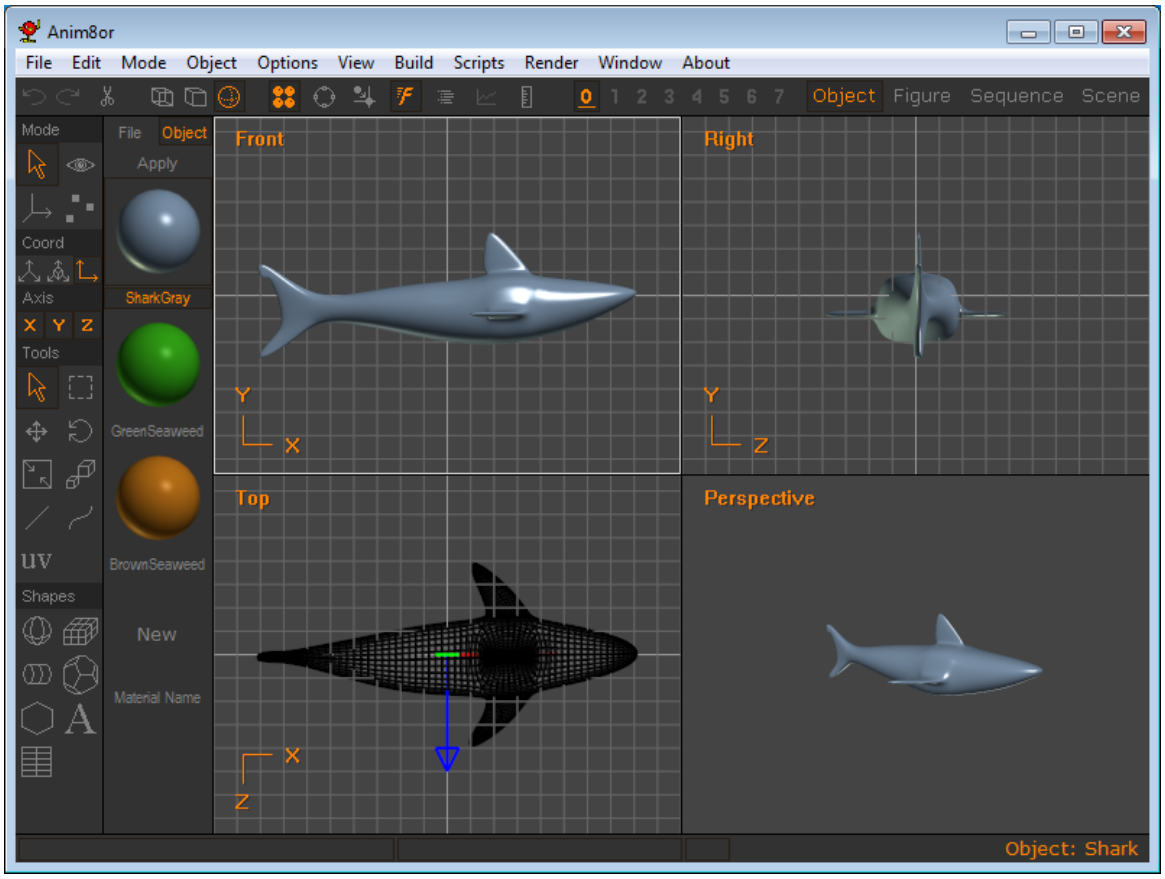
 © 1999 - 2017 R. Steven Glanville

Chapter 3 - Object Editor Basics

Object Editor - Basics & Object/Edit Mode

The **object editor** is where you create and modify new 3D objects. You enter the object editor by default when you run Anim8or. You can enter it at any time by selecting Mode→Object in the menu, or by clicking on the Object tab to the right of the top toolbar.

The basic screen is shown below. On the left is a toolbar. It has icons that represent the most common operations that you will be doing. It allows you to change the mode that you are working in, add basic primitives, etc. You use the menu at the top of the screen to perform less commonly needed actions.



The object editor has four modes: **edit**, **viewpoint**, **axis**, and **point edit**. You change the mode you are currently working in by clicking one of the topmost four buttons on the toolbar. This will also change the lower part of the toolbar to display buttons appropriate for that mode.

Object/Edit mode button. This is the initial mode that the object editor starts in.

Object/Viewpoint mode allows you to pan, scale, rotate, and size-to-fit any or all of your views of your workspace.

© 1999 - 2017 R. Steven Glanville


Object/Axis mode lets you move an object's pivot or origin.


Object/Point Edit mode allows you to manipulate individual or groups of points, edges, and faces.

Object/Edit Tools

In Object/Edit mode there are some buttons just below the six small buttons on the toolbar that you use to manipulate components of your objects. Your actions only affect the selected parts of an object.


You can select something by clicking on it when you are in **select** mode. This button puts you in select mode. You can temporarily switch to select mode at any time by simultaneously pressing the **Ctrl** and **Shift** keys.


You can also select components by dragging a rectangle around them in **drag-select** mode.


These buttons allow you to **move**,


rotate,


scale non-uniformly, and

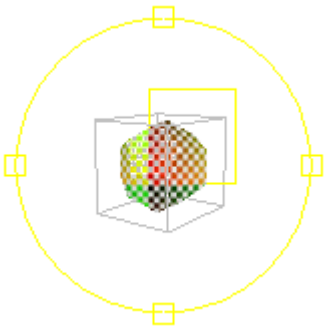

scale the selected components with your mouse.


You can also add and edit **splines** by adding **straight** and


curved segments. Splines are also sometimes called paths since they can be used as a path for motion.

texture coordinate editing tool. You use this tool to interactively apply texture coordinates to selected objects. It adds an overlay to the screen that is similar to the arc-rotate tool used for changing your viewpoint but for textures except that it's used for assigning texture coordinates to objects.

The yellow square represents a texture's basic size. You can move the texture around on your objects with the right mouse button, and you can rotate its orientation with then left button. The middle button scales textures.



Basic Objects

You can add new objects to Anim8or in two ways. The first is to import meshes for the **.3ds**, **.obj**, and **.stl** file formats. You use the Object→Import menu item to do this. The second way is to start

© 1999 - 2017 R. Steven Glanville

with one or more built in **primitives** or shapes. The lower part of the Object/Edit toolbar has several icons for adding new primitives.

Sphere [***S***] - This button allows you add spheres to an object. First click on it. Then move the mouse to where you want the sphere to be added. Click and drag to create a sphere of the size you desire.

If you want to add several spheres, use the right mouse button. Using the left button returns to select mode after adding a single object.

Double clicking on the sphere brings up a dialog where you can change the sphere's properties from the default values.

Cube [***C***] - This button adds cubes and rectangular blocks. You can divide them into as many subdivisions in each axis as you like by double clicking on a block to view its properties dialog.

Cylinder [***Y***] - This button adds cylinders. You can taper them, leave the ends open or closed, and set the number of divisions used to make them in their properties dialog.

Platonic Solids [***M***] - Add various platonic solids and other built-in solids with this button. You can set the current solid type using the Build→Primitives command.

N-Gon [***G***] - This button adds unfilled polygons to an object. They are built as an editable spline so you can make a lot different shapes starting with the right polygon. Splines can be filled, extruded, lathed, edited, etc.

To set the number of sides, use the Build→Primitives→N-gon command.

Text [***T***] - You create true type text objects with this button. Just click somewhere in a window and start typing. If you want to change the properties, double click on the object. You can use any font and style.

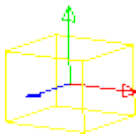
Each letter is a single spline (possibly with multiple parts, like the hole in an O), and the whole string of characters is grouped together. You can extrude text objects directly using the Build→Extrude command.

You can also convert a text object into a group of general-purpose splines using the Build→Convert-to-Spline command.

Warp Modifier [***W***] - This is the modifier button. Use it to add modifiers, which can bend, stretch, warp, and twist another object into many new shapes. Modifiers are covered more fully in a later section of this chapter.

Each object in Anim8or has its own coordinate system. Its pivot is located at its origin. When you scale or rotate an object it is done using the pivot. When you are in wireframe mode, the pivots of selected objects are shown as red-green-blue axes like this:

 © 1999 - 2017 R. Steven Glanville

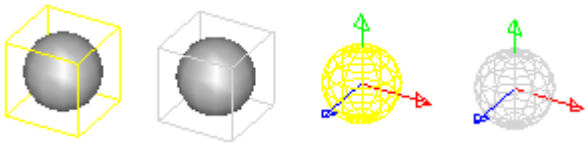


You can move and rotate pivots by using Object/Axis mode.

Mesh vs. Parametric Components

There are two kinds of components in an Anim8or object: **parametric** and **mesh**. Parametric components are defined by a small set of numeric values. These values are used to build a viewable mesh whenever they are displayed. You can edit these parameters by double clicking on a parametric component. Changing parameters will generate a new mesh for display. You cannot modify the mesh directly. Some parametric components are the sphere and cylinders described above.

Meshes however can be fully edited, smoothed, etc., even at the point or face level. You can convert parametric components into meshes with the Build→Convert-to-Mesh command. It is easy to see whether an object is a mesh or not by selecting it. In a filled view, selected parametric components are enclosed in **YELLOW** bounding box and selected meshes in **WHITE** ones. In a wireframe view, parametric components are shown in yellow lines and meshes are shown in white or some other color lines. The four spheres below show the same geometry. The first and third are in their original parametric form, while the second and fourth have been converted into meshes.



Once a parametric component has been converted into a mesh it cannot be reverted.

© 1999 - 2017 R. Steven Glanville

Object Materials

You can view an object's materials by displaying the **material toolbar** with the Options→Materials [***Ctrl-M***] command. Each named material is shown applied to a small sphere. Double clicking on the sphere brings up the material editor dialog allowing you to change that material's properties.

The selected material is shown as a depressed button with a white name. You can set the basic material for any selected component by clicking the **Apply** button.

The **Object** button shows the current object's materials. These materials can be used only by the current Object.

The **File** button shows the project's global materials, those that are shared among all objects in an entire project. You should add materials that are common to several different objects to the global list to help manage them easier.

If you want to define a new material, double click on the **New** button at the bottom of the list.

If there are more materials than can be shown, you can left click in the material toolbar and drag up or down to **scroll** the list.

Materials are discussed in more detail in *Chapter 9, Materials*.

Plug-in Shapes

You can add new parametric shapes to Anim8or beyond those that come as part of Anim8or. These are defined by special script files that you add to the Script Directory. They behave like ordinary parametric shapes and add a new button to the bottom of the left toolbar.

This is the default button for a shape plug-in if the script doesn't define one. You can set a plug-in shapes's parameters in the properties dialog by double clicking on it. You may also be able to scale the size or other parameters with the scale and scale non-uniformly tools, depending on how the plug-in is defined.

More often there will be a button definition as part of a shape plug-in. This one is for a spring. An example of a spring is shown below. The right spring has been converted to a subdivision shape.

See *Chapter 10 Scripts* for more information on adding plug-in shapes to Anim8or.

 © 1999 - 2017 R. Steven Glanville

Splines

You use the curved path  and straight path  buttons to manipulate splines. To add a new spline, you must first deselect everything. Then select either the curved or straight path button and click and drag to place the first segment. The spline will be drawn with a white pick box at its root and a red one at its head like this:

The red box also indicates that the spline is extendable. You can left click on it and drag to add more segments:

Once you have built all of the segments you need, go back to select mode and double click on the spline to bring up its property dialog and uncheck the **extendable** check box. This prevents the accidental extension of your spline when you are trying to modify it in some other way. Your spline will look like this:

Now if you reselect either path button you can click on **knots** and display their control points:

You can drag the knots to new locations, and change the positions of the control points to alter the shape the spline. Double clicking on a knot will show the knot dialog where you can change the knot into either a **smooth** knot or a **corner** knot.

Double clicking on a segment will highlight that segment and let you set the number of straight lines used to draw it:

True Type Fonts

When you click on the text button  you can add text to an object. Click in a view window and a **text** cursor will appear. Type some text and it will appear as an outline in the window. Double click on the text and the text dialog appears. You can change the **font**, make it **bold** or **italic**, and set its **size**. You can always resize a text object but this isn't always the same as changing its initial size with the font properties dialog. Larger fonts are sometimes generated with more detail.

 © 1999 - 2017 R. Steven Glanville

You can also fill and extrude text. Use the Build→Fill and Build→Extrude commands:

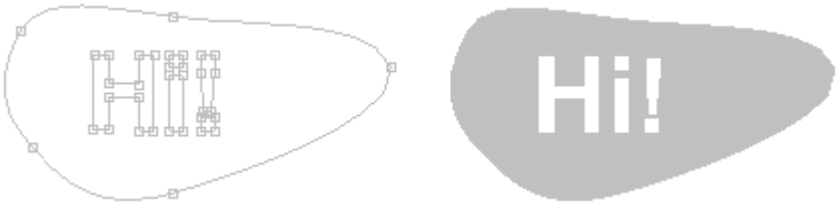
When they're filled or extruded, text objects are converted into a group of meshes. You can do anything to them that you can to an ordinary meshs.

Filling

You can convert any closed spline into a flat, **filled** mesh. Just select that spline and apply the fill operator using the Build→Fill command. The spline should not cross over itself or it will not fill. To fill a spline, simply select it and the select the Build→Fill command.

You can also build complex splines consisting of multiple independent curves. Interior closed areas are filled as cutouts. To build complex splines, select several basic splines and apply the Build→Join-Splines command. For example:

Since text items are actually a group of (possibly) complex splines, text can also be combined with splines and filled. First select the text and convert it to splines with the Build→Convert-to-Spline command. Then ungroup the string of characters with Build→Ungroup. Finally select the spline along with all of the individual characters and combine them into a single multiple spline with Build→Join-Splines. Then fill the result:



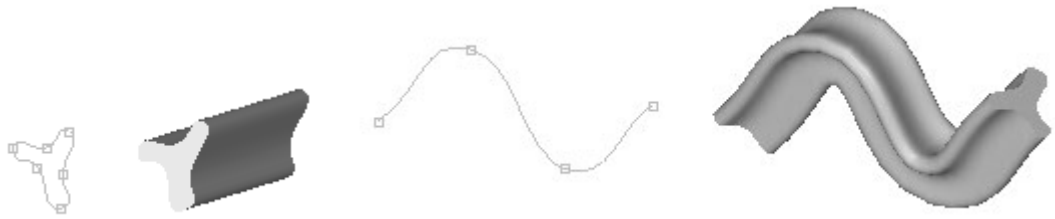
Extrusion

Two more operations you can do on splines are **extrusion** and **lathing**. Both create a three-dimensional mesh out of a spline, but each in a different way. An extrusion sweeps the edges of the spline along a path (straight or curved) while lathing spins them around an axis.

To extrude a spline simply select it and choose the Build→Extrude command. This will display a dialog with several parameters that you can set, such as whether to cap the ends and which

 © 1999 - 2017 R. Steven Glanville

direction to extrude. One option is to extrude along another spline. When you choose this option, you will be prompted to select the spline to extrude along after you leave the dialog. Some extrusions are shown below:



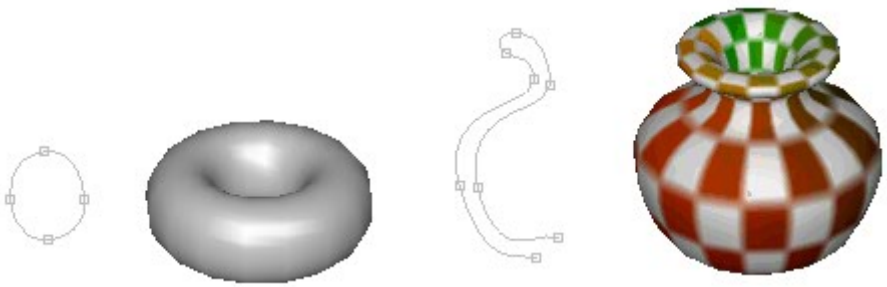
The top left "Y" shaped spline is shown extruded along the Z-axis, and along the lower sine-wave shaped spline.

Note: The "center point" for the extrusion is the initial origin of your spline. Make sure that you create the spline around the origin or the results may not be what you expect.

You can also extrude text. Just select it and apply Build→Extrude:

Lathing

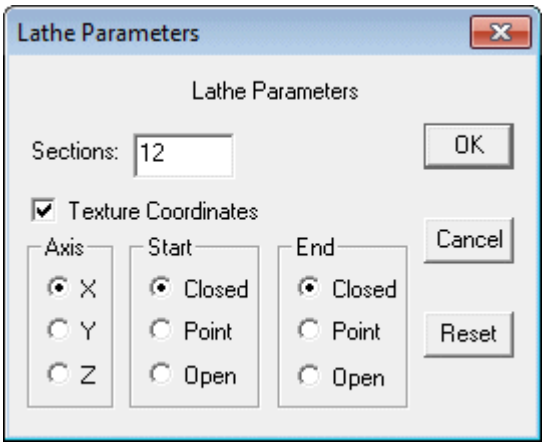
Lathing creates a solid surface from a spline by spinning it around an axis and converting the area it sweeps into a mesh surface. It can produce a variety of different objects depending on the shape of the spline that you lathe. If you lathe a closed spline you can create wheels and donuts. If you lathe an open spline, you can build wine glasses, pots, and vases:



Since you can apply texture coordinates at the same time that you are lathing a spline, it is easy to build things with detailed colorings.

You use the Build→Lathe command to lathe a spline. This displays the dialog:

© 1999 - 2017 R. Steven Glanville

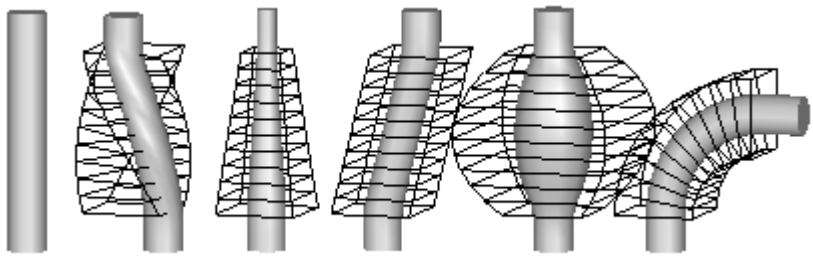


The *Axis* section selects which axis will be used to do the rotation. You can control the behavior of the end points of an open spline with the settings in the *Start* and *End* sections. The three objects below are all lathed from the same spline, but the starting point (the one at the top, not the red one) is handled differently in each case. *Open* leaves the points at the beginning and end in their original place. *Closed* adds a new point on the axis at the same level as the end point and connects to all the end points creating a flat top or bottom. *Point* moves the spline's end point precisely to the axis and then lathes the spline.

***Open**, **Closed**, and **Point** modes used at the Start*

Modifiers

Modifiers allow you to stretch, bend, twist, and warp meshes into an endless variety of alternate shapes. To use them, create a modifier object and **bind** it to an existing mesh with the Build→Modifiers→Bind command. You can double click on the modifier to bring up its property dialog and change its behavior. The effect it has on the target mesh is shown in your object views. When you are satisfied with the shape, select the Build→Modifiers→Effect command and the mesh will be transformed into its new shape.

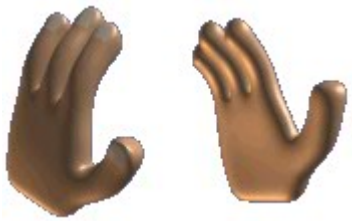


Here is an example of a cylinder modified by a **twist**, **taper**, **skew**, **swell**, and a **bend** modifier. Modifiers preserve texture coordinates and other properties bound to points on the object.

© 1999 - 2017 R. Steven Glanville

Mirror Image

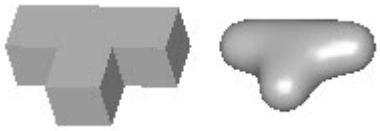
You can also make a **mirror image** of an object using the Build→Mirror-Image command. This builds a mirror image of the original object, a duplicate with the points reflected to the opposite side of the X, Y, or X-axis. These hands are simple mirrors of each other:



If you are building a symmetric object you can save time by building half of it, mirroring that half, and joining the two together.

Smoothing

You can use **smoothing** to round the corners of basic shapes and to give more detail to the model. Anim8or can subdivide any mesh. Just select the objects that you want subdivided, and select Build→Smooth-Object from the menu. Each time you do this you will round the corners a bit more, and multiply the number of facets by four. So be careful, or you'll create a very large model to manipulate!



*Note: Smoothing used to be called **Subdivide Faces** in earlier versions of Anim8or.*

There is a modeling technique called **box modeling** that you can use to model complex objects that begins with a simple box. You bevel, extrude, resize, and warp a few flat faces until you make a rough form of your desired object. Then you apply subdivision, and, viola! You have a nice model. The following were made in Anim8or using box modeling:

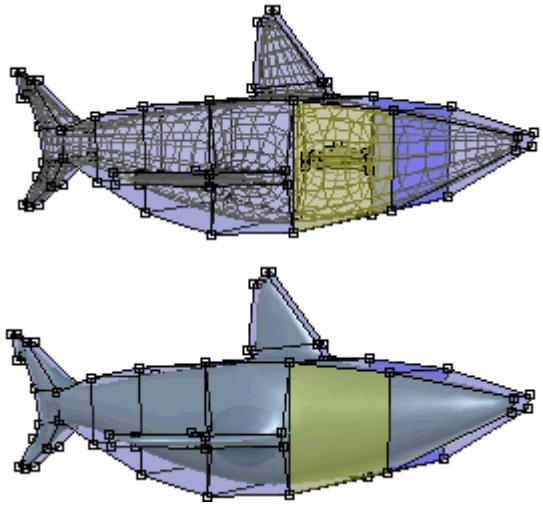
Not a bad hand considering it was modeled by 24 simple cubes! You can learn more about building the box shapes in *Chapter 4: Object/Point Editor*.

Subdivision Objects

You can convert a Mesh into a **subdivision object** with the Build→Convert-to-Subdivided command. This allows you to edit Subdivision objects just like regular meshes, and they are

© 1999 - 2017 R. Steven Glanville

automatically subdivided as you work. This is often an easier way to model than to continually apply subdivision operations to try and see what your model will eventually look like. Selected faces are shown as transparent windows revealing a further refined model inside:



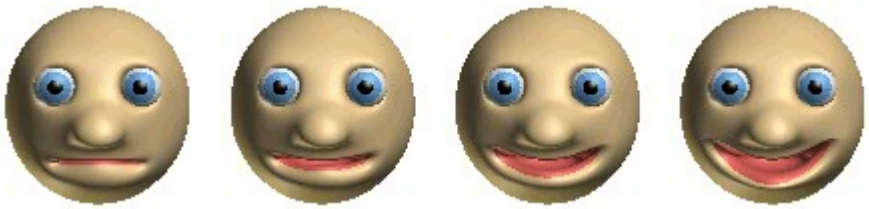
When you resize or extrude a face the internal smoothed view changes shape to follow. (You only see this view in the Point Editor. All other places your model will be shown as fully subdivided images.)

You can convert a Subdivision object into a Mesh with the Build→Convert-to-Mesh command. It will be converted at the subdivision level that you are using to view it in the working views.

Another very important advantage to Subdivision objects is size. Subdivision objects are stored in the undivided format that is many times smaller than the final fully divided model. You can control the level of subdivision used on the screen and in the final image by double clicking and editing the properties dialog.

Morph Targets

A **morph target** is a deformed version of an object where some or all of its vertices are moved to alter the shape. For example you may have a basic model of a character's head with a neutral expression. If you want to animate your character opening and closing its mouth you'd create a morph target with the character's mouth open. You can then smoothly animate between the base shape with a closed mouth and the morph target with an open mouth using a single controller. As you increase the value of the controller the character's mouth opens.



The face on the left has a neutral expression. The face on the right is a morph target created for the face so that it's smiling. The faces in-between are morphs with a value of 1/3 and 2/3.

© 1999 - 2017 R. Steven Glanville

To create a morph target, you first build a base model. It's important to put the finishing touches on your model first since altering the base model can corrupt any morph targets you have added. You then make a morph using the Build→Morph-Targets→New command. Make sure to give it a meaningful name such as "smile" so you can find it when you are animating in the Scene editor.

Next edit your model in the Point Editor by moving vertices to alter the shape. When you're finished with the morph use the Build→Morph-Targets→None command. to return to the base shape or continue to add more morphs.

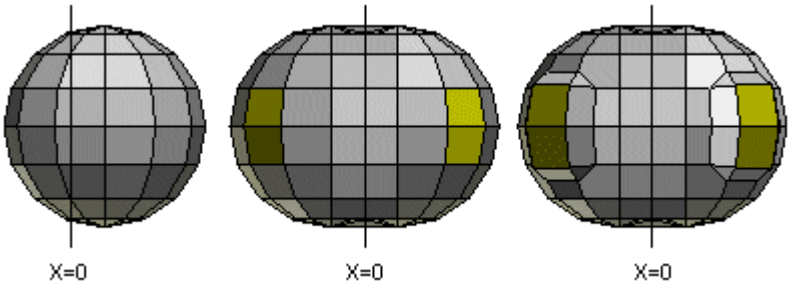
Note: If you use a Subdivision object for your model you will have fewer points to manipulate.

You're now ready to animate you object in the Scene editor.

Continuously Mirrored Meshes

Mirroring is a mesh property that helps you build symmetrical models. This property causes any change to the right side of a mesh to be applied identically to the left, and visa-versa. You can apply mirroring to any mesh with the Build→Mirroring→Convert-to-Mirrored command.

When you initially convert a mesh into a mirrored mesh, only points to the right of the Y-Z plane are mirrored, those with positive X coordinates. All points to the left of the Y-Z plane are deleted.



Once mirroring is enabled you can move, rotate and scale your model like any other model, without the need to keep it aligned to the Y-Z plane.

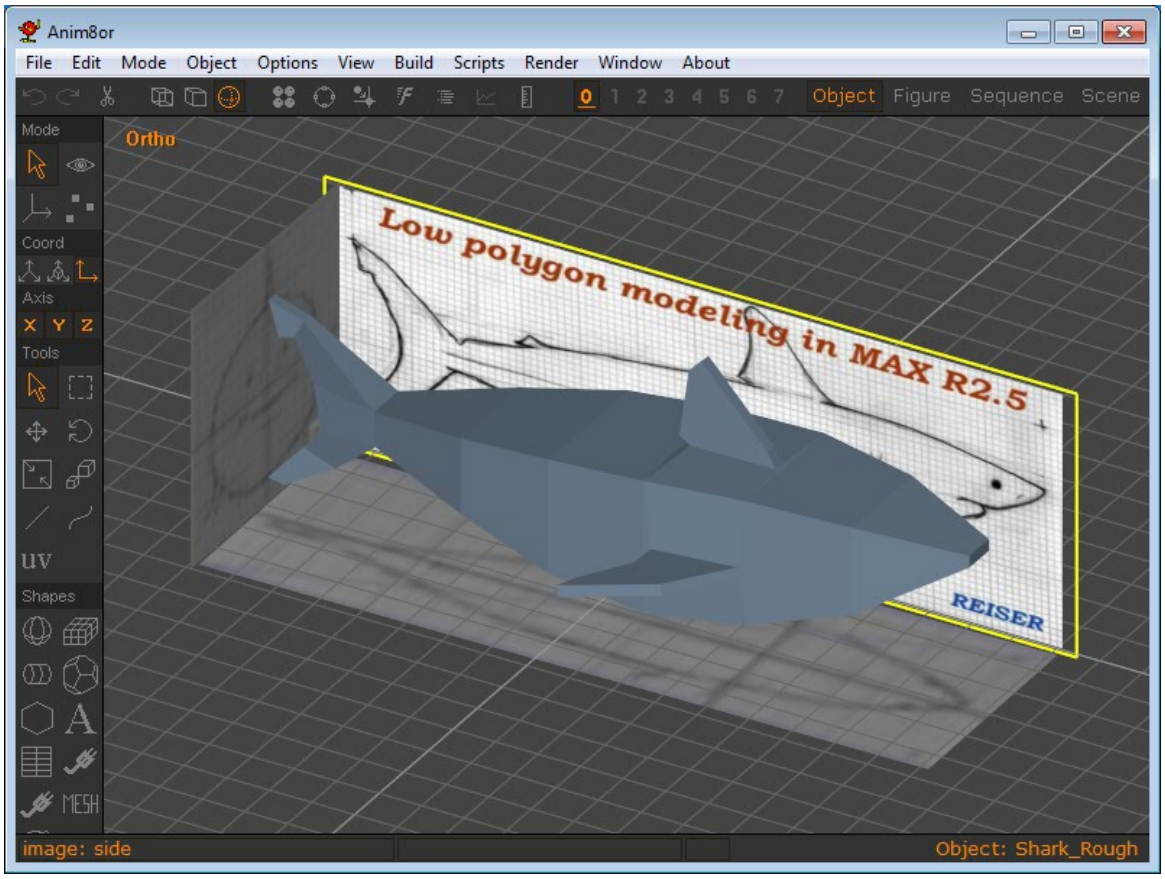
When you are finished with your model, or if you want to start making changes to only one side, you can disable mirroring with the Build→Mirroring→Stop-Mirroring command. If you re-enable mirroring you will lose any asymmetrical aspects. It's a good idea to save a copy of your mesh at the end of the mirroring phase in the event that you want to refine the mirrored aspects.

Note: At this time continuously mirrored meshes do not work with Morph Targets. You first must disable mirroring to start adding morph targets to your models.

Reference Image

You can add **Reference Images** to an object to use as a guide when modeling with the Build→ReferenceCommand command. References are added parallel to the current view, so if you one when you are in the front, top and left views you can model in all 3 dimensions directly over the images. You may need to move them deeper behind your model so that they aren't blocking the view of your model.

© 1999 - 2017 R. Steven Glanville



 © 1999 - 2017 R. Steven Glanville

Chapter 4 - Object Editor Point Editor

Object Editor - Object/Point Mode

The Object/Point Edit button [***P***] on the toolbar changes Anim8or into Object/Point editing mode. Here you can modify individual points, edges and faces of any mesh object. You can move, scale, rotate, add and delete them. You can extrude, twist and scale faces, weld close points together, etc. When you enter Object/Point Edit mode you will usually want to change the display into wireframe with the View→Wire [***Ctrl-W***] command.

Object/Point Operations

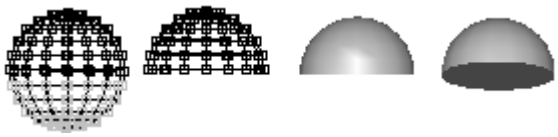
Just below the coordinate system button row there is a new row of three buttons: . You use these to change the selection mode to **point**, **edge** or **face** in order to select different parts of your model. Left-click on these to select just one mode. Right click to toggle individual buttons. The select and drag select modes will then select the corresponding items.

There are two buttons that you will find helpful when selecting points, edges and faces. They control whether you can select **front facing** Front [***[***] and **back facing** Back [***]***] faces, edges and points. By default both kinds are enabled. Click on the buttons to toggle the setting.

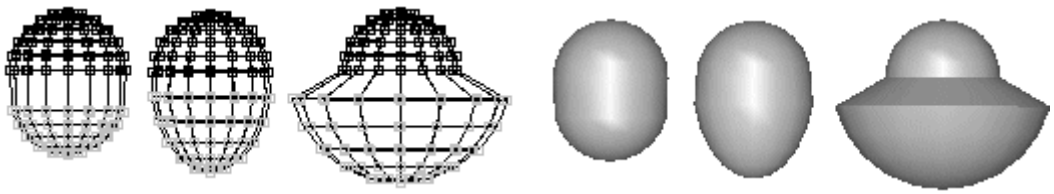
Point Editing

You use **point select** mode [***p***] to modify the shape of an object, and to add and delete unwanted parts of an object. You can manipulate selected points by using any of the move, rotate, and scale buttons. The scale and rotate buttons operate of groups of connected edges, scaling and rotating each group around its center.

If you delete points using the scissors button [***Ctrl-X***], or the delete key [***Del***], any edges and faces that contain those points are also deleted. For example, to delete the bottom of a sphere, first select all of the points below but not including the equator using drag select mode, and delete them



If you move these same points, scale them non-uniformly, or scale them, you can make objects like these:



© 1999 - 2017 R. Steven Glanville

You can use the *Arrow Keys* on the keyboard to make tiny moves, rotations, and scales to selected points as well when the corresponding toolbar button is enabled.

In move mode, click-dragging with the middle mouse button (or alt-right) will move individual points along their normals instead of relative to the screen.

Edge Editing

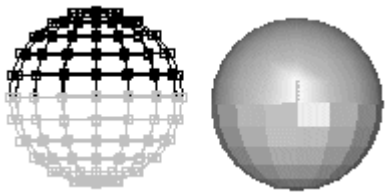
You use **edge select** [**e**] mode  in a similar manner to point select mode. When you select an edge, both of its end points are also selected. Thus all operations from point select mode are enabled as described above.

You can move, rotate and scale selected edges with the normal buttons. The scale and rotate buttons operate of groups of connected edges, scaling and ro-tating each group around its center.

If you delete using the scissors in edge mode, however, only the selected edges and any faces that contain them are deleted. All points are saved. This allows you to change the tessellation of an object more easily, without having to add back previously deleted points.

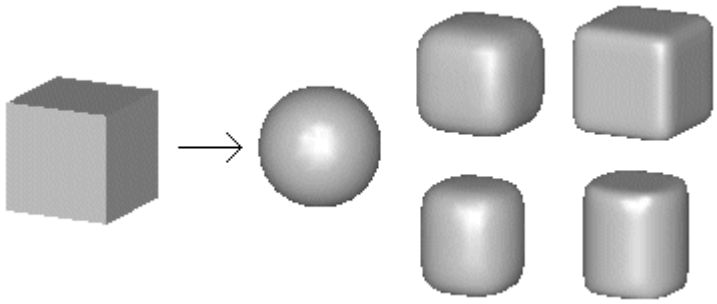
You can also set various **edge properties** for selected edges. Select the edges you want to effect, and then select the Edit→Edge-Properties command. The default setting for edges is **smooth**. This means that if the solid angle defined by the two faces is less than a certain threshold, the edge will be smooth. Otherwise it will be creased. You can change the value of this angle by double clicking on a mesh when in Object/Edit mode to bring up its parameter dialog.

You can mark an edge as **creased**, so that it won't be drawn as a smooth surface but will appear as a sharp crease. This does not change the geometry of the model in any way; just the way normals are computed so that it appears smooth or creased. The following sphere has the lower half of its edges set to creased:



Another property that you can set in this dialog is how **rounded** an edge will be after it has been smoothed. Normally when a mesh is smoothed, all edges become smoother. Each application of smoothing increases the number of faces in a mesh, and smooths all edges. You can mark any edge so that it will not be smoothed, but will retain a sharp crease, for one or more cycles of smoothing. The following shows a cube and several possible smoothings of it:

© 1999 - 2017 R. Steven Glanville

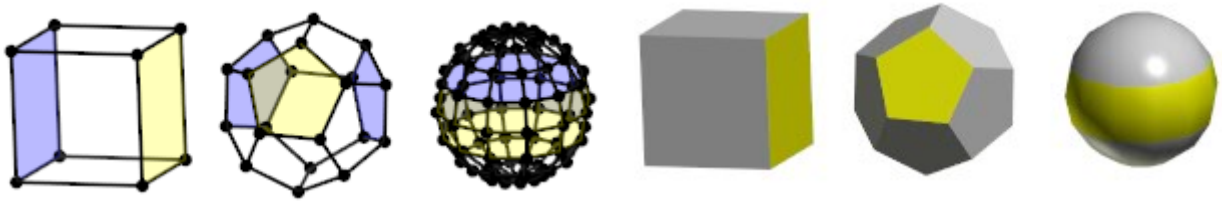


The sphere-like ball is the result of a simple cube being smoothed several times. The top two rounded cubes are the result of smoothing a cube with all of its edges set to a rounded level of 1 and 2. The bottom two rounded cylinders are from a cube with only the top 4 and bottom 4 edges set to a rounding level of 1 and 2.

Face Editing

You use **face select** [***g***] mode  to select and manipulate the faces of a mesh. You can delete selected faces with the scissors button or delete key, but points and edges remain unchanged. You can move, rotate and scale selected faces with the normal buttons. The scale and rotate buttons operate of groups of connected faces, scaling and ro-tating each group around its center

Selected faces are shown as filled even in wire frame mode. The front side is shaded **YELLOW** and the backside **BLUE** to help you see what faces are selected. This is also the case in filled mode, but you usually only see the yellow front sides.



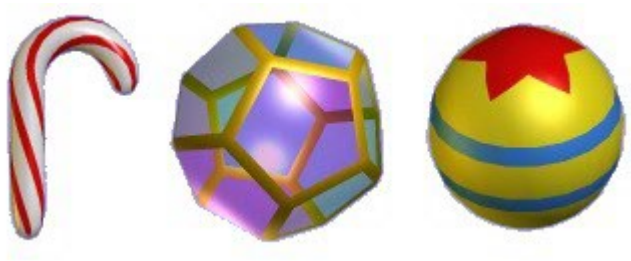
Applying Multiple Materials

You can also apply different material properties to individual faces. Simply select the faces that you want to change, display the material toolbar (use Options→Materials [***Ctrl-M***] if it isn't already visible), select the material to apply, and click the apply button 

in the toolbar.

You can use as many materials on a mesh as you want. With a little imagination you can make objects like these:

© 1999 - 2017 R. Steven Glanville



The UV texture tool is also available in the Point Editor. You first select the faces that you want to texture, and then use the UV tool to interactively apply the coordinates. It helps if you already have a textured material applied to those faces.

A Note on Selecting Faces

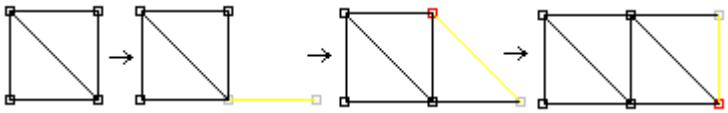
It can sometimes be rather tedious to select the exact faces that you want, especially in a complex model. There are several tricks that can help out.

The **select** mode selects faces that are facing the viewer over those facing the other way. Often changing the view of an object, or rotating the view with the arc-rotate tool, so that different faces are in front, can help. This is one reason that selected faces are shown in **YELLOW** even in normal views where you normally can't modify individual faces.

The **drag select** mode only selects faces that are entirely within the selection rectangle. You can often add faces one-by-one until you've selected what you need. The front/back buttons can be useful here as well.

Adding Points and Edges

You can add new points and edges to a model with the Add Edge button in the toolbar. To add edges to an existing object, click on one of its points and drag the mouse out. A new edge and point will be added where you release the mouse. If you move the mouse to an existing point the end of the edge will snap to the existing point, thus creating an edge between the two points. You can build complex meshes quickly using this method:



If you want to add a point that is close to another one but not make it the same point, hold down the *Shift* key *after* you click the mouse button down and hold it until you release the mouse.

If you want to start a **new** mesh, press the *Shift* key *before* you press the mouse button. A new mesh will be created with a single edge and 2 points.

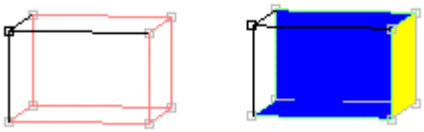
Note: This command only adds Points and Edges, not Faces. You can use the Edit→Fill-Holes [***J***] command described next to add new faces.

If you enable **grid snap** [***Ctrl-G***] when you add points you can make objects with exact measurements more easily.

 © 1999 - 2017 R. Steven Glanville

Adding Faces

Once you have added new edges to a mesh, you will be able to fill in the gaps with new faces. You do this by first selecting all of the edges that surround where your new face or faces will go. Then you use the Edit→Fill-Holes command to fill them.



There are a few things to note before you start adding faces:

- Faces can only be added to completely connected loops of edges. If there are any gaps, no face will be added.
- All edges in the loop must *not* already be part of two (or more) faces. Only edges that are part of *at most one face* will be considered. ou can tell how many edges an edge participates in by its color when selected:

Color of Edge	Number of Edges
RED	None
GREEN	One
WHITE	Two
VIOLET	Three or More

- You can select multiple edges and do a fill of the whole batch at once. Anim8or will start by filling any selected edge loops of length 3. Then it will fill those of length 4, etc. until no more faces can be added. This will sometimes create undesirable results and you will have to try again in smaller batches.
- You may have to apply the Edit→Filp-Normals [***N***] and the Edit→Fix-Normals [***X***] commands to orient new faces properly after they are added.
- You may have to delete some faces in your mesh before you can add new ones.

Connecting Meshes

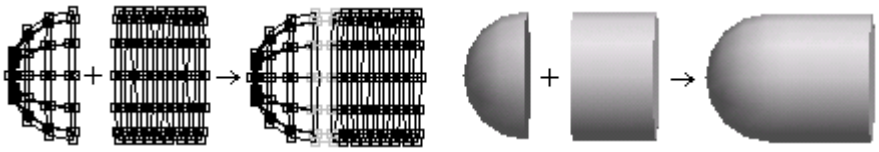
You can attach two separate meshes to create a single, connected mesh using the JoinSolids, Add Points/Edges and Add Faces commands.

- First position the two meshes with the parts that you want to attach near each other. Leave a small gap for a row of new faces that will be used to connect them.
- In the Object Editor, select both meshes and apply the Build→Join-Solids command. This doesn't add any new connecting faces but combines them into a single Mesh allowing you to do so.

 © 1999 - 2017 R. Steven Glanville

- Next in the Point Editor, add new edges between the pairs of points that you want to attach.

- Select all of the edges that you just added, and those on the new boundary.
- Use the Edit→Fill-Holes [***J***] command to add the connecting faces.



Merging Points

You can merge separate points into a single point with the Edit→Merge-Points [***L***] command. First select the points that you want merged. It is best if they are very close to each other to begin with. Then apply the Edit→Merge-Points command. A dialog appears where you can enter the maximum distance allowed between pairs of points. Click OK and all eligible pairs will be merged.

Connecting Meshes (2)

You can also use this technique to connect two meshes. It works best if the connecting point sets are similar in size and number. For example connecting half of a Sphere that has 12 longitude lines with a cylinder with 12 sections around is a snap.

Connecting Meshes (3)

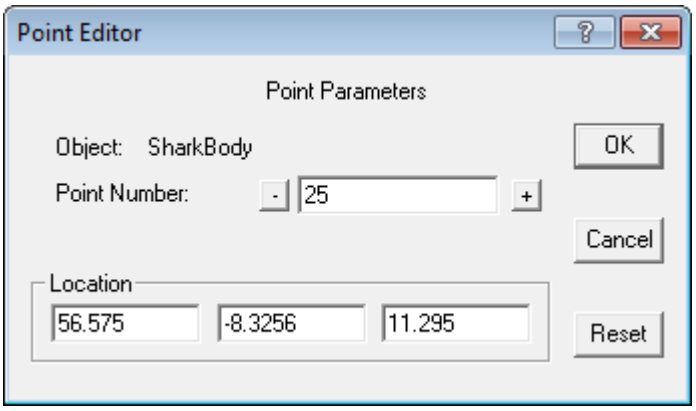
You can often build one half of a symmetrical object, then mirror it with Build→Mirror and join the two halves into a single mesh with Build→Join-Solids Then position the two halves close together, select the adjacent points and use Edit→Merge-Points in the Point Editor to make a seamless connection. This merged object was then smoothed one time to round off the corners.

© 1999 - 2017 R. Steven Glanville

Point and Line Parameters

When you are in Point Select mode you can double click on a point to bring up a dialog box that lets you read and alter its position numerically. This is very useful when you want to precisely position a part of a mesh when building a mechanical model.

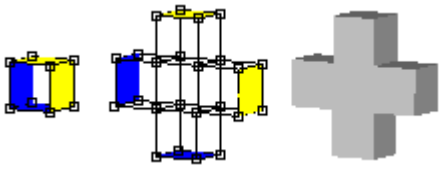
When you are in Edge Select or Face Select mode you can similarly double click on an edge to edit its properties.



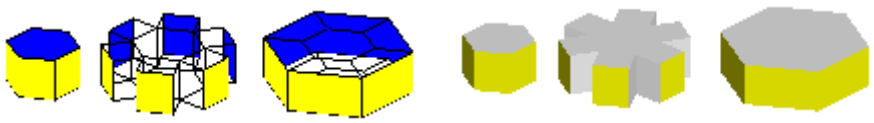
Face Extrusion and Manipulation Tools

You can manipulate selected faces in a variety of ways using the lower group of buttons in the toolbar. They can be used to build very complex models from a small number of primitive shapes using the box modeling method.

The **extrude** [***X***] tool extrudes all of the selected faces. Each face is translated in the direction that it faces, either out or in, and new polygons are added to connect the face to its original location.

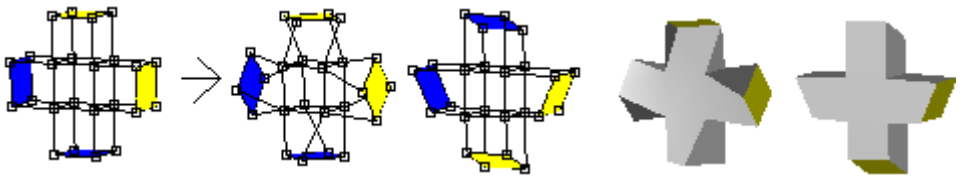


Similar to the extrude tool, the **extrude-connected** [***Y***] tool extrudes selected faces in a Mesh. However adjacent selected faces remain connected. New faces are only added to the edges between a selected and an unselected face.



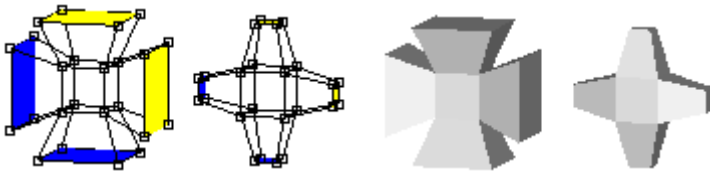
In the image above the 6 faces on the first solid are shown extruded and extrude-connected.

The usual rotate button **rotates** [***R***] each selected face around its center. If a group of adjacent faces are selected the entire group is rotated around the group's center.

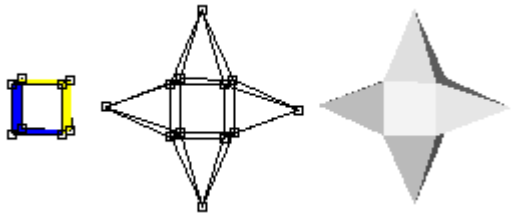


© 1999 - 2017 R. Steven Glanville

You can **scale** faces both **uniformly** [***s***] and **non-uniformly** [***n***] using the usual scale buttons. As above, adjacent selected faces are scaled as a group.

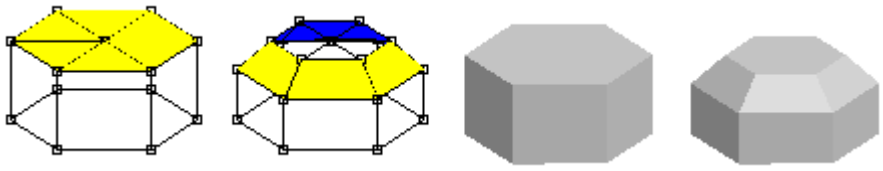


This one will replace faces with a **peak** [***P***].

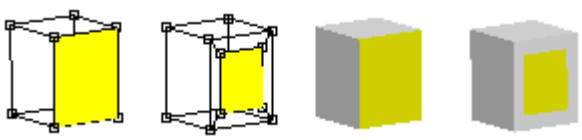


The **bevel** [***b***] tool is useful for adding beveled corners to surfaces. You can bevel faces, edges and points of any mesh. Selected edges and points are turned into new faces, cutting the corner off of the original model as shown below:

Selected faces are handled as a group. Edges between a selected face and an unselected face are beveled but edges between two adjacent selected faces are left alone.



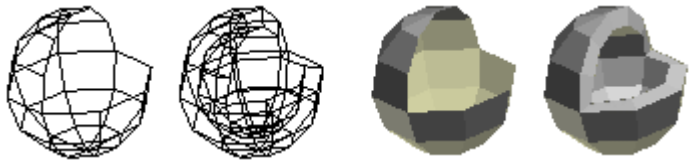
The **inset** [***I***] tool is also useful for adding faces to surfaces. Each selected face is replaced by a smaller version of itself and new faces that connect it to the original edges. This is an easy way to add more detail to a particular part of a model, or to make a hole for a door, window, or eyes.



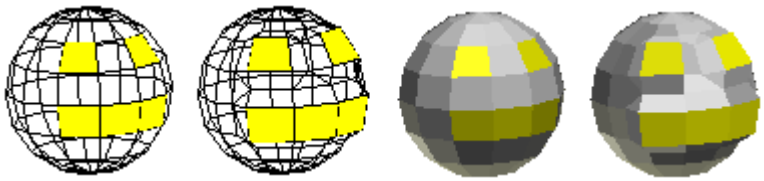
The **shell** [***u***] tool adds thickness to the walls of a Mesh. To make a shell simply click on a

© 1999 - 2017 R. Steven Glanville

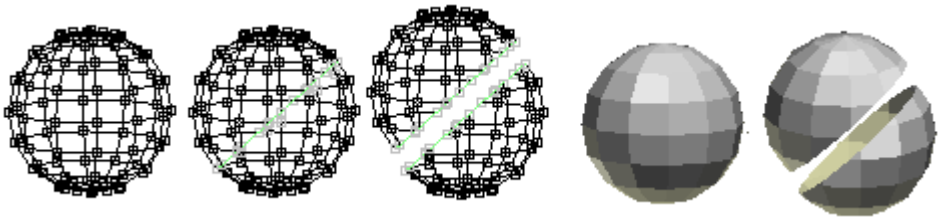
Mesh or a Subdivision shape and drag the mouse. The basic Mesh cannot be completely closed. It must have at least one "missing" face.



The **bump** [***B***] tool raises all the selected faces up on a bump.

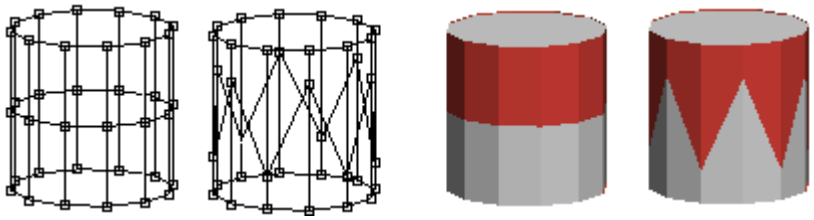


If you want to cut a slice through a model you'll find the **cut faces** [***C***] tool quite handy. It slices through edges and faces when you click-drag the mouse across an object. If you slice all the way across an object as shown below you can then use the Edit→Loop-Cut command.



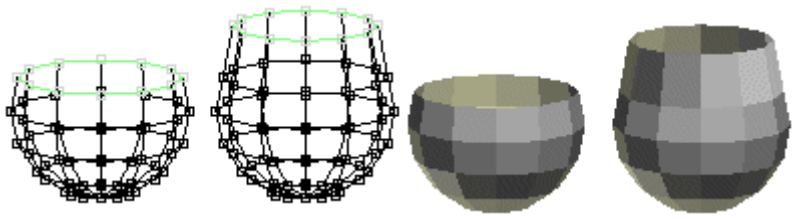
You can divide a face into two by clicking on one vertex and dragging to another. The connecting line will snap to a vertex when you click nearby. Or you can disable the snapping if you press and hold the **Shift** key when applying a cut. You may want to disable the back facing edges from being cut as well by turning off the Back button.

The **slide** [***S***] tool lets you slide a point along an edge, or extend the edge. Just click on a point and drag the mouse in the direction of one of the edges that enters it. This will select the active edge and shorten it. If you want to lengthen it instead just keep dragging but in the opposite direction.

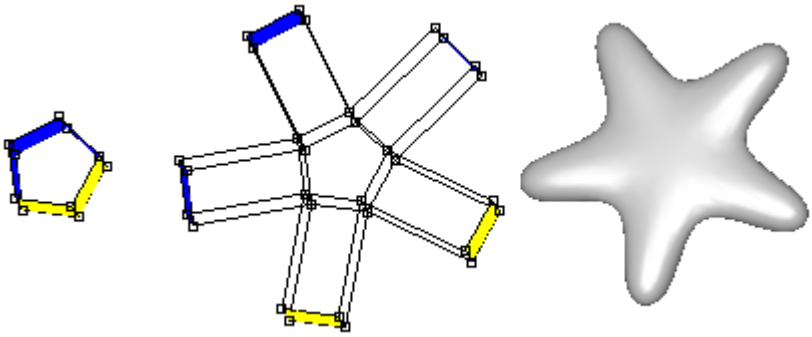


The **edge extrude** [***E***] tool extends your object for all selected edges that have only one face, adding a new adjacent face.

© 1999 - 2017 R. Steven Glanville



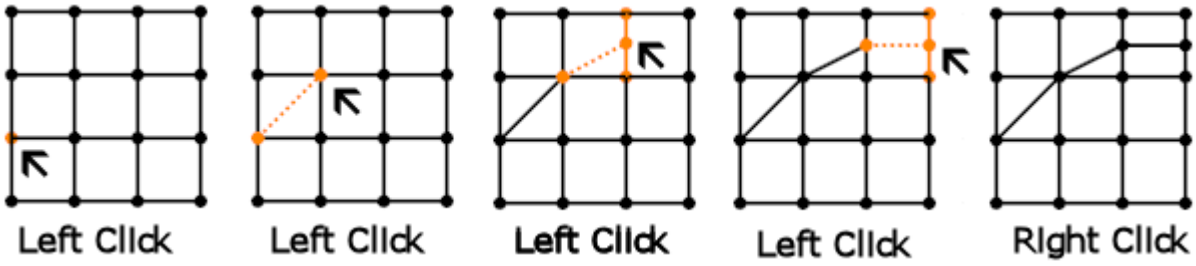
When you use these face manipulation operations you may wish to convert your mesh into a **subdivision mesh** with the Build→Convert-to-Subdivided command in order to build more complex shapes. For example, start with an extruded pentagon, select all 5 outer faces, extrude them, and convert to subdivided to make a starfish:



Alternately you can use one or more applications of the Build→Smooth-Object command for the same effect.

Topographical Knife

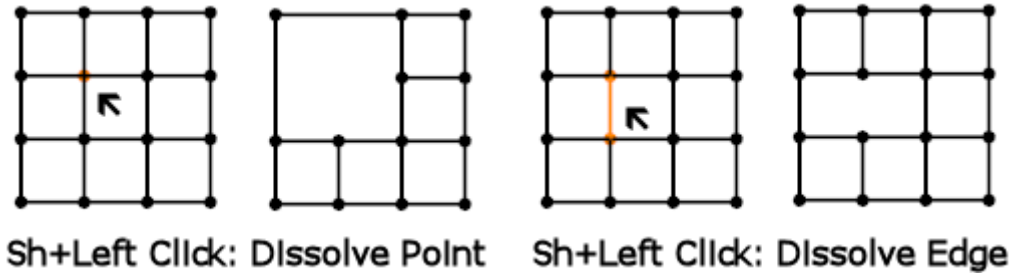
The **topographical knife** or topo [***k***] tool is a major addition to version 1.0. This is an extremely versatile tool that allow for quick editing of meshes in multiple ways.



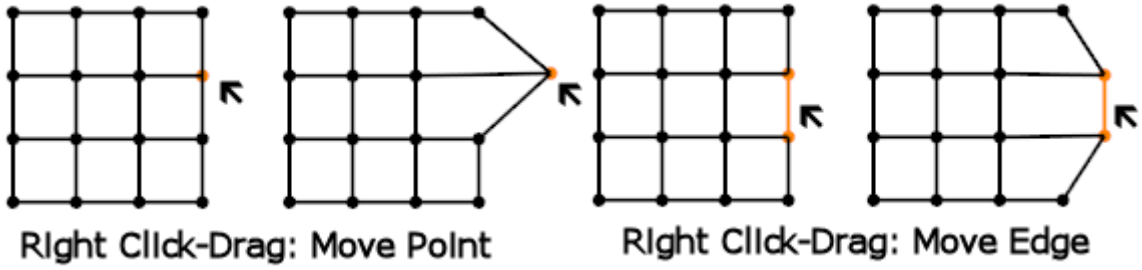
Split Faces: LEFT-click on a point or edge begins adding edges to split faces giving them more detail. Release the button and move to another edge or point on the same face. Click again to divide the face in to two. You can continue adding a sequence of connected edges in this manner. RIGHT-click to exit this mode.

Add Edges: Ctrl + LEFT-click allows you to add edges arbitrarily, not just across a single face. This includes adding edges connecting across gaps and isolated points not in any face. As long as you keep pressing **Ctrl** you can add multiple, arbitrary edges. Releasing **Ctrl** reverts to the original behavior. Again use RIGHT-click to exit.

© 1999 - 2017 R. Steven Glanville



Dissolve: Shift + LEFT-click on a point or edge to remove or "dissolve" it. Your mesh will reform and stay connected in a simpler form. This is a useful way to clean up areas of your model with too much detail or awkward geometry.



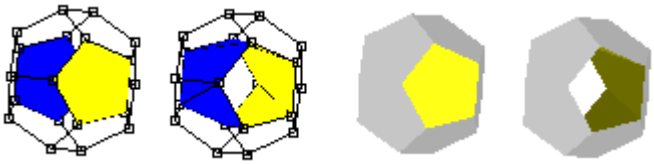
Move Edges and Points: RIGHT-click on a point or an edge and DRAG to move that point or edge along with the mouse. Anim8or uses the active coordinate system for the move, so make sure you've selected the one you want.

Move by Normal: Shift + RIGHT-click and DRAG moves in the direction of the point's normal.

Face Editing Commands

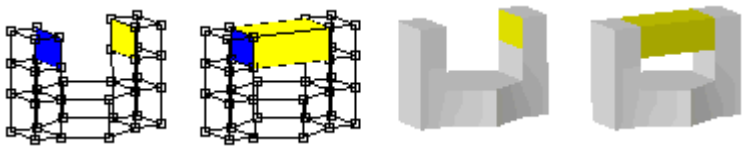
Anim8or has several other useful face editing commands. You will find them under the Edit menu in the Point Editor.

The **bridge** [**=**] command allows you to do several useful operations. To use it you first select two different faces. They have to have the same number of edges but can belong to different Meshes if you want. The Edit→Bridge command will then connect the two selected faces with new faces. You can poke holes through a Mesh:

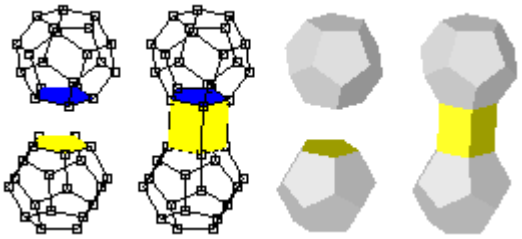


You can connect two parts of a Mesh to form a bridge:

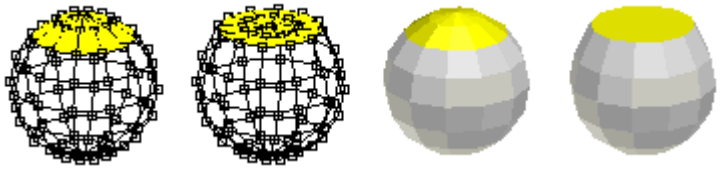
 © 1999 - 2017 R. Steven Glanville



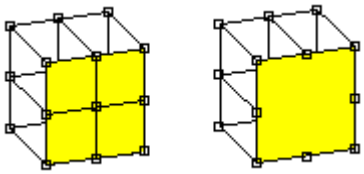
And you can join two different Meshes into a single mesh:



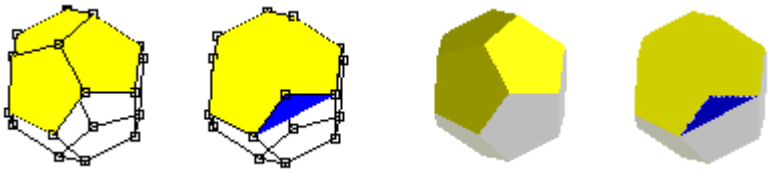
The **flatten** command does just what its name suggests: it flattens groups of selected faces. All the vertices of each group are projected onto a flat plain. You can choose the group's average face direction or the X, Y, or Z-axis. Flattening is quite useful when you need a level base for a model, and can be used with the merge face command to help simplify your models.



The **merge faces** [***M***] command combines groups of selected faces into a single face. You can simplify areas of your models that are too detailed this way.



You have to be careful to keep the faces **convex** and relatively flat or they may not render properly. Convex faces are those that don't have any points inside their outer boundary. Faces with interior points are called **concave** and can cause rendering errors:



In the example shown above three pentagonal faces are merged into one big face with 9 sides. It is concave and doesn't draw properly as a result. It's easy to see in this example because of the stray back facing part that is blue when selected. Other times it may not be as obvious what is wrong.

© 1999 - 2017 R. Steven Glanville

If you are working with a Subdivision object this is not as important. The faces that you are editing aren't drawn in your final object. Subdivision will smooth the mesh making it much better formed along the way.

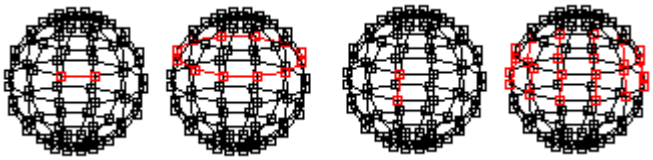
Miscellaneous Commands

There are several miscellaneous commands in the Point Editor that you will find useful. One group helps you make certain useful of selections that could otherwise be quite tedious. The other group modifies objects in various ways.

Selection Commands

A **quad loop** is a chain of adjacent quads, or 4 sided faces, that are attached across opposite sides. The **quad loop select** [***q***] command extends selected edges along quad loops as long as there are exactly four faces joined together at each vertex. For models like the sphere below this results in a closed ring around a latitude line. For vertical edges it only extends to the poles since there are more than four faces sharing the pole points.

The **quad ring select** [***Q***] command works on chains of adjacent quads, or 4 sided faces. When a quad has a selected edge, the opposite edge is also selected. This is repeated until all quads in a "ring" are selected, or until a face without 4 faces is encountered. (Selections are shown in red here to make them better visible against a white background.)



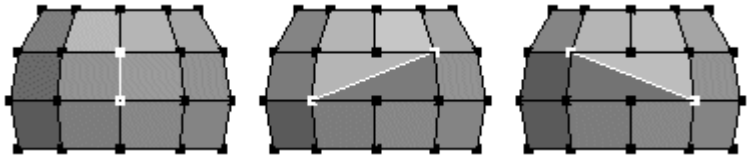
You will find these commands under the Edit→Select menu.

Editing Commands

Connect Edges [***j***]	adds new edges between the centers of any selected edges that share a face.
Cut Edges [***Z***]	splits each selected edge into two in the middle.
Detatch Faces [***D***]	removes all selected faces from their mesh and adds them to a new mesh.
Subdivide Faces [***U***]	divides those faces into several smaller faces. It does this by adding a new point in the center of each selected face and connecting it to the center of the edges of that face.
Loop Cut [***l***]	splits a mesh into multiple parts along closed loops of selected edges.
Spin Quads [***/***]	Spin quads is a useful tool for modeling. It changes which corners of a face are connected by rotating an edge to adjacent points. The faces adjacent to

© 1999 - 2017 R. Steven Glanville

the selected edges must have 4 sides, hence the name.



These commands are under the Edit menu.

Invert Selection [*i*]	selects all unselected points, edges and faces and de-selects those currently selected.
Grow Face Selection [*G*]	selects any unselected faces that are adjacent to selected ones.
Select Connected [*c*]	selects all points that are in a connected region of the mesh to at least one selected point, all edges that are in a connected region to selected edges, and all faces connected to a selected face.
Select by Material	allows you to select all faces with a given material.
Select Malformed Edges	selects any edges that are in 3 or more faces. These edges can have unpredictable side-effects when you edit or render.
Select Orphans	selects any points and edges that are not currently being used in a face. This can help you remove unnecessary geometry from your models.

These commands are under the Edit→Select menu.

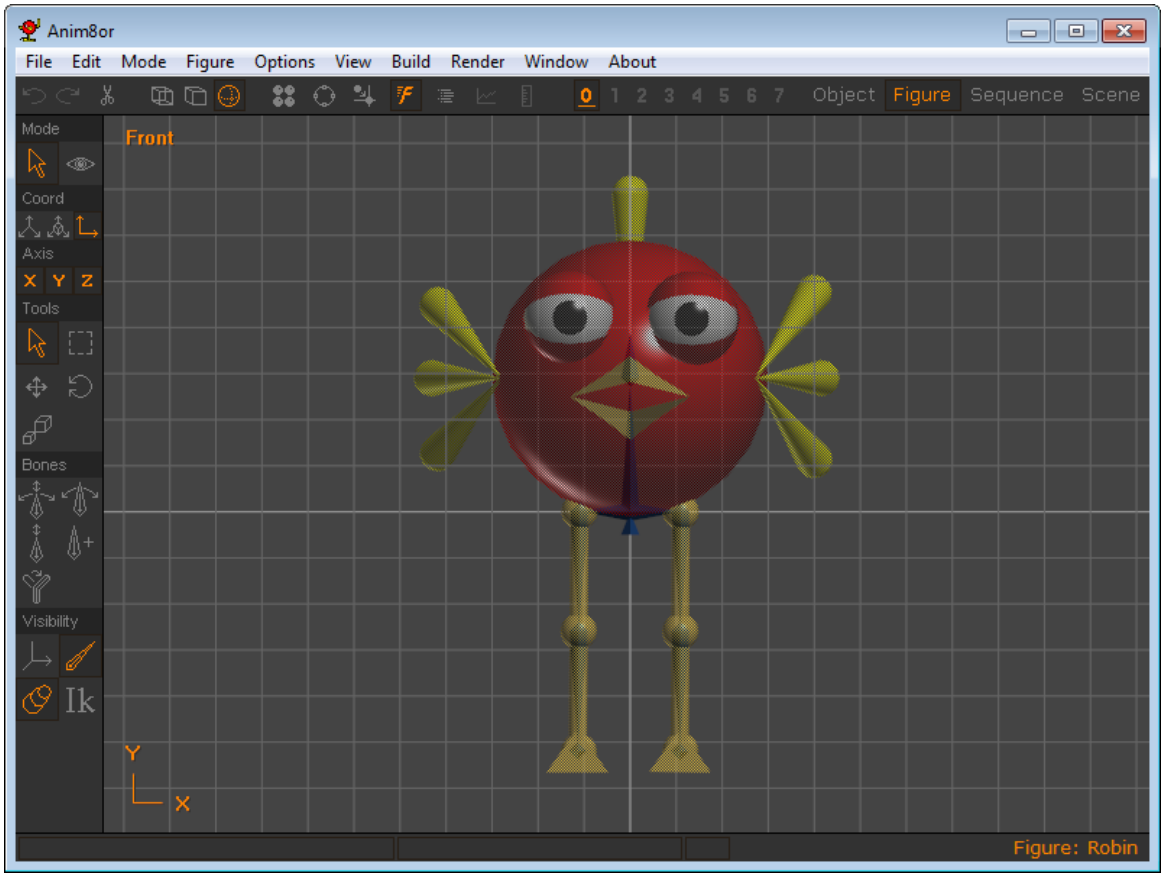
© 1999 - 2017 R. Steven Glanville

Chapter 5 - Figure Editor

Figure Editor

The **figure editor** is where you design animated characters to put into Anim8or's scenes. The main difference between a figure and an ordinary object is that a figure contains a **bone** structure, or skeleton, that you use to bend and shape it in different poses. By setting the pose of a character in a few **key frames** of an animation, you can bring it to life. Anim8or will smoothly fill in all the in-between poses by bending the joints of the skeleton just the right amount. You can enter the figure editor at any time by selecting Mode→Figure in the menu, or by clicking on Figure in the right part of the top toolbar.

The basic screen is similar to the Object Editor and is shown below. You can also view multiple viewpoints simultaneously, as with the other editors.



Notice the bones that are faintly visible inside of the character's body and legs. You can control the visibility of bones, your character's body parts, and other aspects to help with the design. This character doesn't have any particularly interesting bones in his body, but does have animatable legs and feet.

The mode section in the figure editor has two modes: **edit** and **viewpoint**. You click on one of the icons to switch between these two modes.

© 1999 - 2017 R. Steven Glanville

Figure/Edit [***A***] mode button. This is the initial mode that the figure editor starts in. You use it to do all of your character design.

Figure/Viewpoint[***V***] mode allows you to pan, scale, rotate, and size-to-fit any or all of your views of your workspace.

Figure Basics

There are two different parts that you make when building a figure: the **body**, which is all of the visible parts of your character that will appear in the final images and the **bones** or **skeleton** which you use to pose and animate your character.

You will usually design all of your character's body parts in the Object Editor as different objects. When you add them to your character you are only making a reference to the original object. This way you can add an object several times without increasing the size of your working files, and you can make changes to all the instances by editing a single object.

There are also a couple of "built in" shapes in the figure editor to help with preliminary design and motion tests. The "ball and stick" legs shown here are made from built-in cylinders and balls.

Once you have made your character's parts, you can switch to the Figure Editor and build a skeleton. Starting at the initial root bone, add bones one at a time to make the movable structure of your character. For example, if you want human-like legs, you need to add a bone for each of the thighs, lower legs, and feet. You can give each joint its own range of motion which makes animation a lot easier.

Once the skeleton is designed you can add the character's visible parts. Then link each part to a specific bone, and when that bone moves, so does the body part. You can also move, scale and rotate each part individually within the coordinate space of its bone.

Figure/Edit Operations

You use the first group of buttons to select various parts of your character, and to place the visible parts on the bones in a similar manner to the same buttons in the Object Editor.

You use this mode to **select** [***a***] both your character's bones and body parts. As usual, you can use the right mouse to select additional elements while keeping the original ones selected.

This button sets **drag-select** [***d***] mode. You drag a rectangle on the screen and select all

© 1999 - 2017 R. Steven Glanville

objects that fall within the final area. Again, using the right mouse button adds to the current selection.

The next three buttons apply to the objects that you have attached to the skeleton, not to the bones themselves. You use this one to **move** [***m***] an object around relative to the bone that it's attached to.

You can **rotate** [***r***] objects with this button.

This button **scale**s [***s***] objects. You have to select an object before you can modify it.

You use the next set of buttons to edit the bones of your character's skeleton once it has been built.

The **edit bone** [***E***] tool lets you both rotate a bone and change its length. There are 3 distinct actions:

1. Click-dragging on the **rotation widget**'s controls rotates the bone according to what part of the widget you click but doesn't alter its length,
2. on the tip of **the** bone rotates the bone and alters its length to follow the mouse, and
3. on the bone itself rotates the bone around its X, Y or Z-axis with the Left, Right and Middle mouse buttons.

This mode allows you to **rotate** [***R***] your character's joints into their **nominal** position. This position defines the zero value for all rotations that this joint can perform. It is not the same as the **default** position of the joint, which is where the joint is when it's "relaxed", but is used to define the bone's coordinate system.

You use this button to change the **length** [***L***] of a bone. When it is pushed, you can click on a bone and stretch or shorten it with the mouse. You may click anywhere on a bone but you must first select it if Fast Select [***Ctrl-T***] is not enabled. You can also double click on a bone and set the length and other parameters in a dialog.

You use this mode to **add** [***N***] new bones to a skeleton. To add a new connecting bone, click on the **end** of a bone and drag to set the bone's length. The new bone will align in the direction that you move the mouse.

You can also use the Build→Insert-Bones command to insert a new bone as the parent, and to specify a precise length.

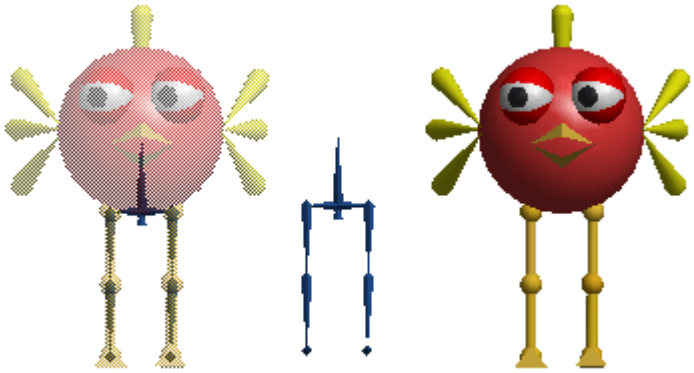
This button allows you to **skin** [***S***] part or all of a skeleton with a single Object. Then when the joints of a bone bend the object deforms and bends seamlessly along with the skeleton. There is more detail on how to use this skinning tool later in this chapter.

© 1999 - 2017 R. Steven Glanville

Visibility

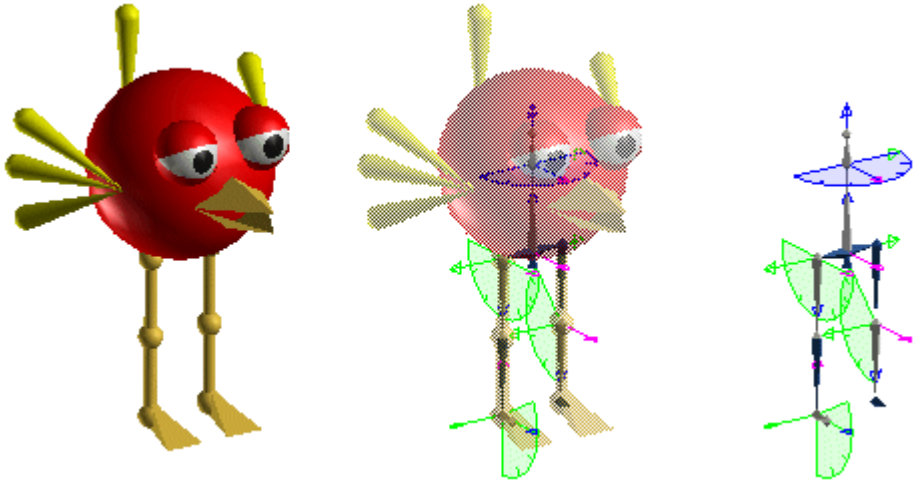
You use the final group of buttons to control what is visible in the edit window. You may find it easier to manipulate your character's skeleton without its body getting in the way, or want to see the final, solid look without the bones showing. These controls will do just that.

You use these two buttons to toggle the visibility of **bones** [***B***] and **objects** [***O***] or body parts. If both of them are enabled then your character's body will be shown in a sort of transparent view with the bones visible inside. You will find this indispensable when arranging your character's parts. If only one is enabled, the corresponding items are shown as solid objects.



This shows a character with a simple skeleton. Only the legs have bones, so you could only animate the legs. The face would be frozen in that silly look unless you add additional bones to animate the eyes, wings, beak, etc.

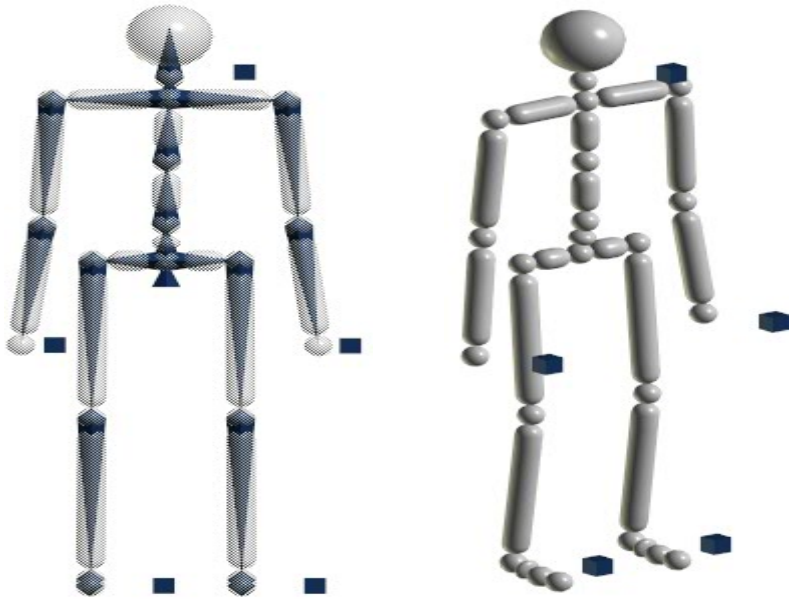
This button toggles a visible **axis** [***X***] for any selected bones. It also shows you the range of movement for their joints.



The **inverse kinematics** [***I***] button displays your figure's IK control handles. You add IK chains by selecting the first and last bones in the chain and using the Build→AddIKChain command. Handles are shown as small blue cubes and are primarily used to select individual chains. You can move and scale them in the figure editor to make individual IK con-

© 1999 - 2017 R. Steven Glanville

trols easier to select. Animation using IK is done in the Sequence and Scene editors.



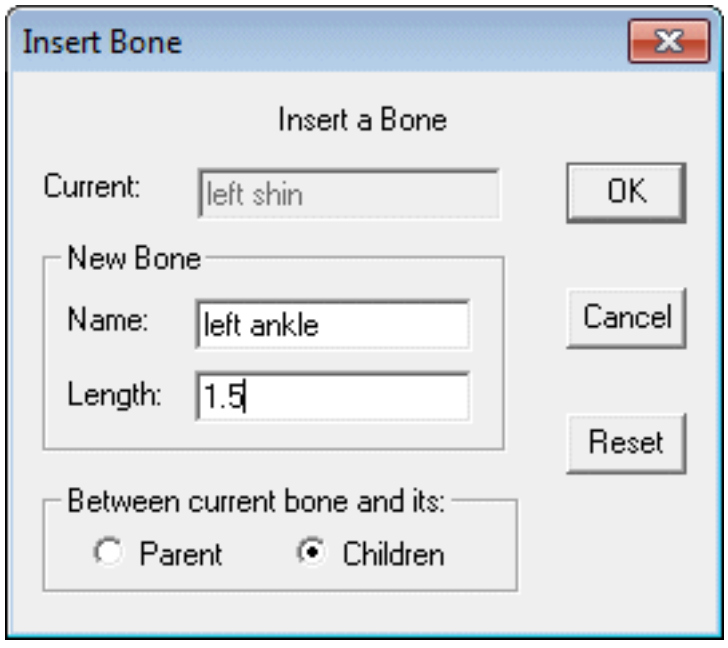
Building a Skeleton

Each figure's skeleton starts with a **root** bone. By adding **child** bones the root, and then to successive bones, you construct a skeleton one bone at a time. Each bone is owned by its **parent**, and any movement that the parent makes also moves all of its children.

When you add a bone it is aligned in the same direction as its parent. So you will normally have to rotate it to its proper orientation.

If you want to **delete** a bone, first select it and then choose Edit→Cut from the menu bar. This will delete the bone and all objects attached to it. Any lower bones will be moved up and attached to the deleted bone's parent.

You can also **insert** a bone into your figure. First select a bone and then use the Build→Insert-Bone command. You can insert the new bone either as the parent or as the child of the selected bone.



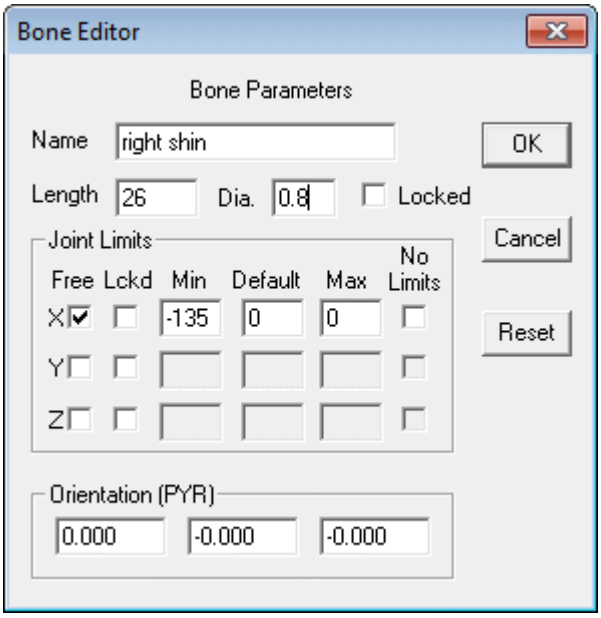
© 1999 - 2017 R. Steven Glanville

This is a simple skeleton. It is made out of 9 bones plus the root. This skeleton is built with two symmetrical hips and legs. The lower two bones in the right leg are selected and shown is white.

The foot has an extra diamond visible on the end. This gives you an easier target to grab onto when you are manipulating the bone. The lower leg has one as well, but it is hidden inside the foot bone and isn't visible.

Flexible Joints

In Anim8or you decide which joints are **flexible** and which ones are frozen in a **fixed** position. By default they are all fixed. You add flexibility in the bone editor dialog that appears when you double click on a bone.



You control the rotation of a joint around each axis independently. Clicking on the "**free**" check box for an axis allows it to rotate around that axis. You can set the minimum, default, and maximum allowable angles as well.

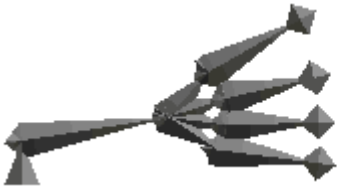
You use the **X** and **Z**-axes to control the bending of a joint, and the **Y**-axis to twist around. Clicking the single "**Locked**" check box will prevent any changes to a bone's length or position in the basic skeleton. It's a good idea to lock your bones once you've got things set up. This prevents their accidental alteration. You can still change all the joint parameters, but since this is done in a dialog instead of with the mouse, it is less likely that you will do something accidentally. The "**Lckd**" check box for each axis is used when you are animating a character and doesn't apply to the figure editor.

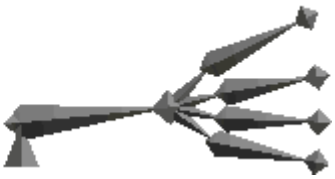
Angle values are limited to values between **-359** and **359** degrees with a maximum difference of 359 degrees.

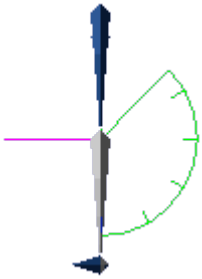
If you check the "**No Limits**" box then you will be able to rotate the bone freely in that axis. This kind of joint is called a free or unconstrained joint because there are no restrictions on its position.

Working with a lot of small bones, such as in a hand, can create a confusing jumble of overlapping bones. You can change the diameter use to show each bone in the Bone Editor by changing the value in the **Dia**. field to help clear up the view.

© 1999 - 2017 R. Steven Glanville







This is a side view of a skeleton. The show axis [***X***] button is set so you can see the joint rotation limits of any selected bones. The x-axis is shown in **green** the y-axis in **blue** and the z-axis in **violet**.

The bone selected here is the lower part of the right leg and forms the right knee. It can only bend in the x-axis and like a human knee can only bend backwards in this case a maximum of 135 degrees backwards.

Adding Body Parts

Once you have built a skeleton, you need to add the visible body parts that will appear in your final images. Each body part is an object that you add to a specific bone's local coordinate system. First select the bone that you want to add your object to, then use the Build→Add-Object menu command to add it. Your object then behaves like that bone. When the bone rotates, so does the object.

You can also add simple cylinders and spheres. They aren't very sophisticated, they just look like sticks and balls, but they do allow you to do quick tests to see how your skeleton behaves. You add them using the Build→Add-Cylinder and Build→Add-Sphere menus.

You can change a few of the properties for these simple shapes as well. Double click on one to bring up its property dialog, and use the Settings→Component→Material command to set their color. These basic shapes are not as flexible as Objects that you build yourself. They have some fairly restrictive limits and aren't meant for any kind of final rendering. For finer control than this you need to build things in the Object Editor.

Skinning

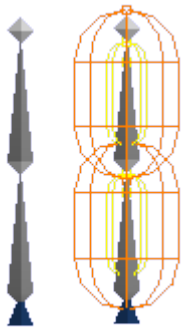
In the previous section you've learned how to add an Object as the offspring of a specific bone. It follows all of its parent bone's movements, but remains a rigid, inflexible Object. This is what you want when making a robot. It's not exactly what a living human or animal does. You want them to be flexible, to change shape and bend along with their skeleton. This is exactly what **skinning** does.

Once you've built an Object and a skeleton, you can attach your object to several bones at the same time with the Skinning tool. Here you set how strongly each bone controls each point on your Object. The stronger the influence the closer it will follow that bone's movement.

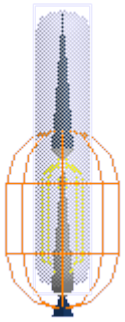
© 1999 - 2017 R. Steven Glanville

There are two ways you can set the level of influence. The first is to use the **influence volume** that surrounds each bone. Any point in a skinned Object that lies within this volume is bound to that bone to a certain degree depending on how close it is to the bone. You can use this influence to animate your object so that it bends along with a Figure's skeleton. Alternately you can paint your own **skinning weights** to bind your Object to a Figure.

Influence Volumes

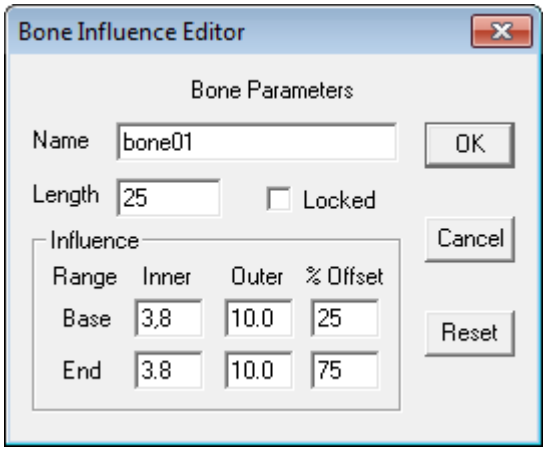


The two bones at the right show their influence volumes or ranges. The inner yellow area is where the bone exerts maximum control. This does not mean that other bones can't also control points that fall within this region, just that the bone is a strong influence. The outer orange area is the limit of the bone's control. That bone does not affect points that lie beyond this boundary at all. The only exception is that if a point lies outside all bones' areas of influence, it is simply controlled by its base bone, the one that it is initially attached to.



To skin an object you first attach it to a bone called its **base** bone. This is the default bone controlling all points in the object but all bones in your figure are capable of influencing the shape of your object. Then enter Skinning mode by pressing the skinning button on the toolbar  [***S***]. Now select your object by clicking on it. This will enable skinning for your object and add its base bone to its influence set those bones which can alter its shape. To add other bones simply click on them and confirm that you want to add them in the dialog that appears.

You can adjust the size of a bone's influence in two ways. You can double click on it and enter numeric values, or you can edit the influence regions directly on the screen.

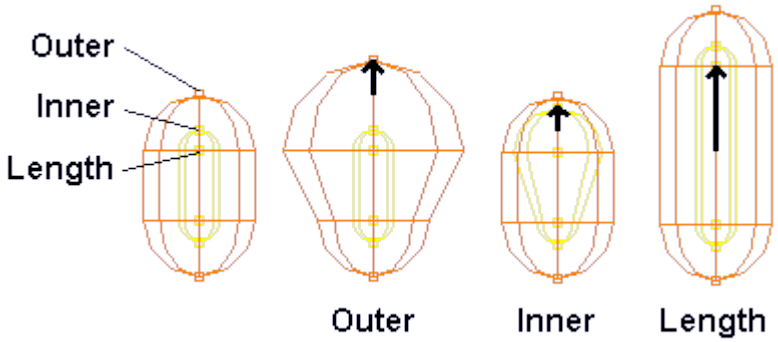


You can set the size if the inner and outer range for each end of the bone. You can also set the position along the bone where they start and end as a percentage of the bone's length.

Changing sizes directly on the screen is often more effective but it can sometimes be tricky to do because things are often cluttered with several overlapping regions.

There are six control points on each bone, three at each end. If you turn off the visibility of both bones and objects by clicking on the buttons at the bottom of the toolbar you will get a clearer picture of what's there. You use the one at the tip of the orange area adjust the size of the outer region, and the one at the tip of the yellow area for the inner one. The inner yellow point adjusts the offset.

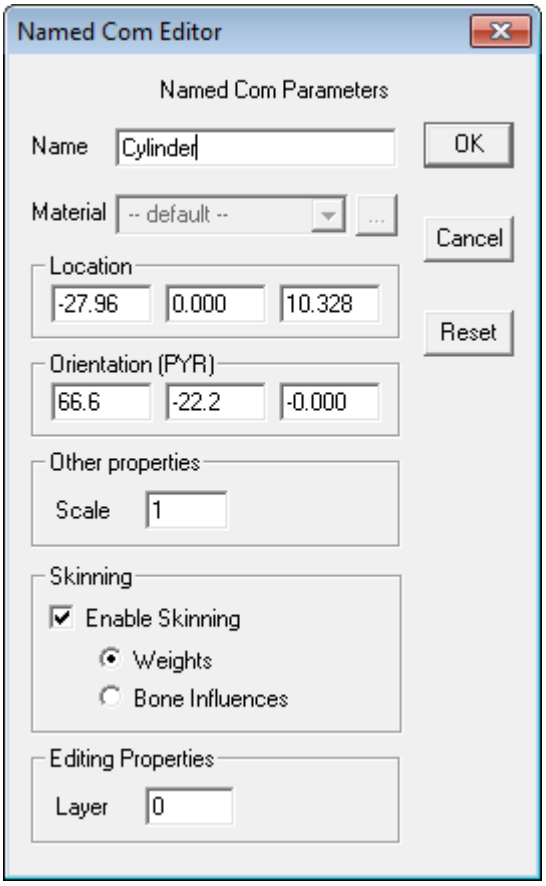
© 1999 - 2017 R. Steven Glanville



One final note: Don't attempt to adjust your bone's positions in the Figure Editor and expect to see your mesh bend! It doesn't work that way. You will only be adjusting the bone's position, and area of influence, relative to your object. You must go to the Sequence or Scene editor to start animating your character.

Skinning Weights

For more control you can also paint skinning weights directly on your objects. It's best to start with a basic influence volume weighting and then convert to using weights. You do this by double clicking on an object and selecting the Weights button in the Skinning area of the parameter dialog.



To paint weights enable skinning  and select an object. The display will show the object's bones in contrasting colors. You then right-click on a bone to paint its weight on the object. The brush is a green circle with a cross in the center. Hold the left mouse but-

 © 1999 - 2017 R. Steven Glanville

ton down and drag the brush across your object and the influence of the bone will be increased for the points that you paint. The color of the object will change to show the new weights.

The center of the brush has the most "paint" and will make the biggest change in weights where it passes. It decreases to the edge where the change stops. You can change to size and strength of the brush with the Build→Weight-Brush command.

Now you're ready to go to the Sequence Editor to make some animated sequences. Then on to the Scene Editor for your final movie!

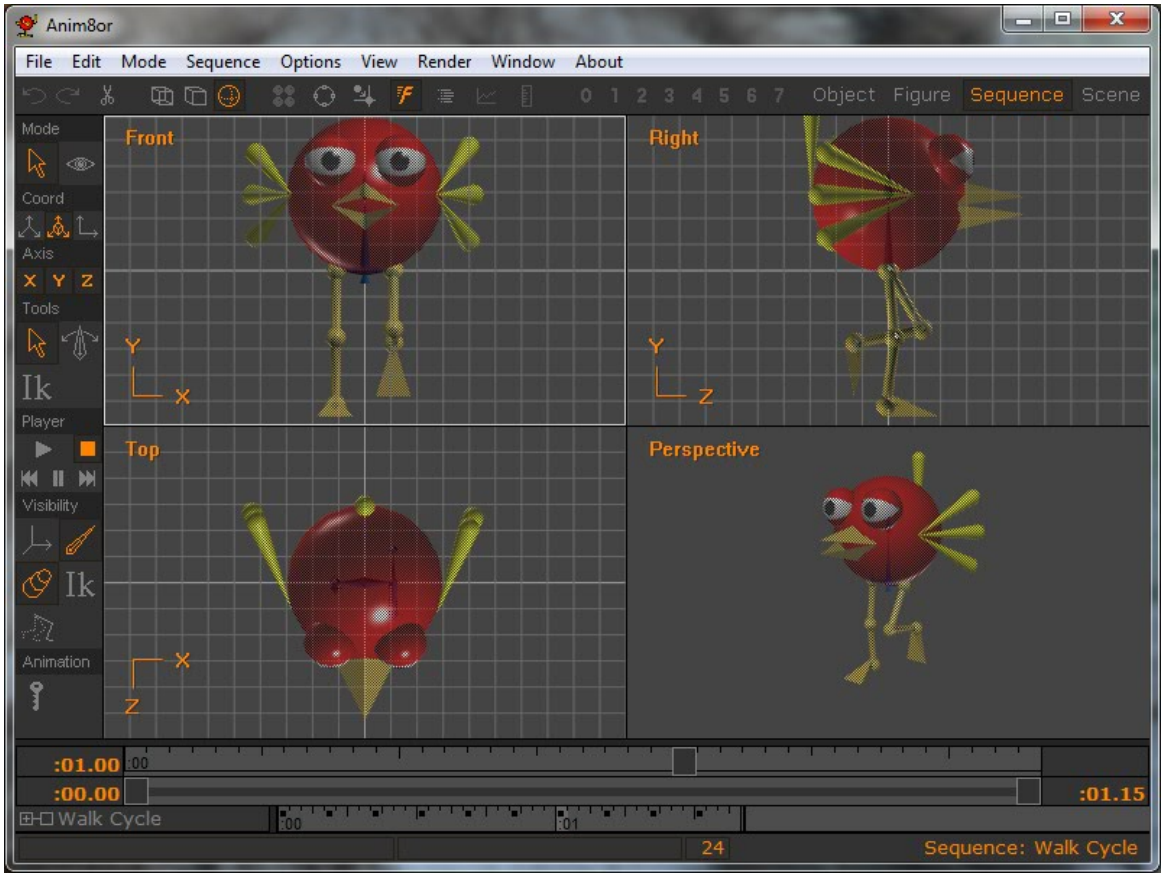
© 1999 - 2017 R. Steven Glanville

Chapter 6 - Sequence Editor

Sequence Editor

The **Sequence Editor** is where you can create movement that gives personality and style to your characters. A **sequence** is a short block of key frame poses, such as one cycle of a walk. The easiest way you can animate a figure is by placing it in a scene and adding one or more sequences.

Longer animations can be built from several short sequences, such a building a long walk out of multiple applications of a walk cycle. Anim8or automatically joins the adjacent sequences into a single, smooth movement, using the key frame positions as guides. Here is a screen shot of the Sequence Editor in action:

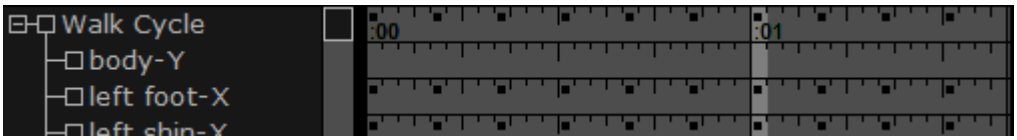


Time Track

The **time track** at the base of the Sequence Editor indicates the current frame shown in the view-ports. The frames that have one or more joints set in key positions are marked with small black squares ▪ in them. The numbers **:01** show the time in seconds since the start of the Sequence of a position on the track. The current frame is highlighted with a lighter gray background.

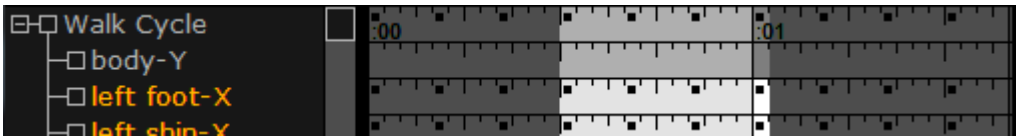
 © 1999 - 2017 R. Steven Glanville

You can expand the track to see a tree view showing details of individual joint's keys by clicking on the plus symbol ⊞ next to the title.



Key positions are shown for each joint/angle with a small black square in the frame. You can also select a joint by clicking on it in the expanded track. Resize the expanded track view by clicking and dragging the vertical resize bar ▬▬ at the bottom of the toolbar up or down.

The track will automatically scroll to keep the current frame visible when a sequence is too long to show it all at once. If you want to see a frame that is off the window, you can either use the Settings→Sequence command to set the frame numerically, or you can use the Left or Right Arrow keys to play the sequence forwards or backwards until you reach your frame.



You can move to a particular frame in a sequence by clicking on that frame in the track. If you click and drag the mouse it will select a range of frames. Clicking on the name of a bone selects that bone and highlights it in the track window. Clicking on the title of the sequence selects all bones. Mouse buttons have their usual meaning on the bone names: the right button adds to the selection, and the middle button subtracts. Use Shift-Click to select a new frame range while still keeping the same current frame.

Scrubber Bar

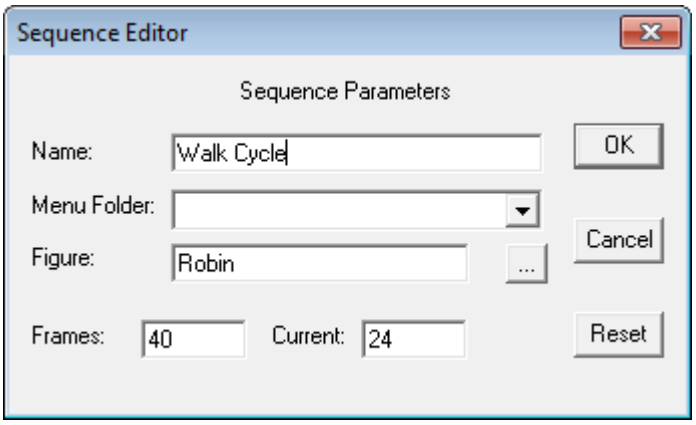
It's often useful to play your animations freely forwards and backwards to check the motion, an action called **scrubbing**. Anim8or's sequence and scene editors have a tool call the **scrubbing bar** that allows you to do this. You can click on the button in it's top row and slowly or rapidly drag the mouse right and left and the animation will run forwards and backwards in response.

The bottom row allows you to set the limits of the scrubbing range. You can click-drag the two end buttons to adjust the range.

Sequence Basics

You can't animate a character before you have one! So it isn't surprising that you must first design a character in the Figure Editor before you can give it any movement. Once you have one, you switch into the Sequence Editor and a new, blank sequence is created. Select the Settings→Sequence command to show the Sequence's settings dialog:

© 1999 - 2017 R. Steven Glanville



Click on the ... button and select your figure from the list. You can also set the number of frames in the sequence in this dialog. Now your figure will appear in the viewports and you can begin to animate it. You can also set the Sequence's length and set the current view to a particular frame with this dialog.

The sequence editor has two modes: **edit** and **viewpoint**. You click on one of the icons to switch between these two modes.

Sequence/Edit [***A***] mode is the initial mode that the sequence editor starts in. You use it to set all of your poses and to view the motion of you've created.

Sequence/Viewpoint [***V***] mode allows you to do the usual pan, scale, rotate, and size-to-fit any or all of your views of your workspace.

Edit Operations

[***a***] You use this button to **select** bones in your characters. The right mouse button allows you to select multiple bones

[***R***] This mode allows you to adjust the **angle** of your character's joints to new positions and for new key frames.

[***I***] In the Tools section of the toolbar, this is the button for **inverse kinematics**, a helpful tool for posing and animating your characters.

Visibility Options

These controls allow you to show different aspects of your character on the screen when you are working.

[***X***] This button toggles the visibility of the **axis** and the range of motion of any selected joints.

[***B***] [***O***] These buttons control the visibility of the **bones** and **body parts** of your character, like in the Figure Editor.

© 1999 - 2017 R. Steven Glanville

[*i*] In the Visibility section this button toggles the visibility of the IK handles when you are not actively editing them.

[*G*] Use this **ghost view** button to show multiple frames of a sequence simultaneously. You can compare previous key poses and in-betweens that Anim8or creates with another frame's to help see how they fit.

Animate Button

These controls allow you to show different aspects of your character on the screen when you are working.

[*K*] This is the **animate** button. It is your magic wand for making things move. You use it to add and change key positions. When you change the angle of a bone you sometimes want to do it for all frames in a sequence, and sometimes just in the current frame. The key button gives you this choice. If it is not pressed changes apply to the entire sequence. If it is pressed only the current frame is changed. The next sections explain this in more detail.

What is a Key?

If you had to set the position of *each* joint of your figures in *each* frame of an animation, that's about all you'd ever do! Fortunately there is an easier way to animate called **key frames**. With it you pose your character in a few frames that represent what it should be doing in that particular frame, and the computer smoothly fills in the in-between frames. You get the best results, naturally, if the frames you choose represent the significant points in your character's movement. That's why they're called "key".

There is no reason that all the joints have to be a key in the same frame. If your character is walking forward and slowly turning his head you may want dozens of key positions in the legs but only a couple in the head turn.

Making a Key Pose

When you first create a Sequence, your character is shown in its default position. You can change its position by bending its joints. But you can only bend the ones that you made flexible in the Figure Editor, and only in the ways that you said they could bend.

© 1999 - 2017 R. Steven Glanville

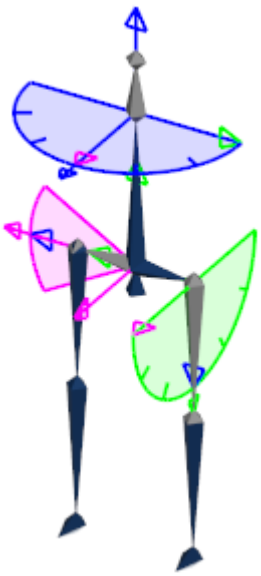
To make a key pose enter Rotate Bone [***R***] mode, then enable Animation [***K***] . Select the joint or joints that you want to move, and then you can either use the trackball or rotate the bone directly to pose your character.

Using the Trackball

The most recent selected bone will display a trackball centered on its starting joint. The axes that are free to rotate are shown in color on the trackball. Click-drag on the axis that you want to rotate to add a key position for that joint.

If an axis isn't highlighted in color it can't be rotated either because it isn't enabled or it's too difficult to edit from that angle.

Click-dragging on Bones



Click-drag on a bone to rotate it directly. Anim8or will hide the trackball and display a transparent arc that shows the limits of rotation possible.

Each mouse button rotates around a different axis.

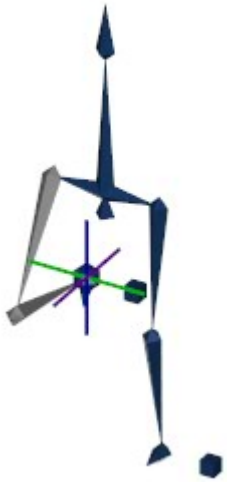
To rotate around the X-axis, which is shown in **green**, use the left mouse button. The joint will try to follow the mouse.

To rotate around the Y axis shown in **blue** use the right button. Since the Y-axis rolls the joint around the bone's length, and doesn't change the direction it's pointing, you have to move the mouse to the left or right to move the joint.

To rotate around the Z-axis shown in **violet**, use the middle mouse button.

Click on the tip of the bone to rotate it in a more general way.

Inverse Kinematics



If you added any IK chains to your figure in the Figure Editor, you can key all of its bones in a single action. Enable IK mode [***i***] then click on the IK chain's handle, a small blue cube, to display the IK widget and the End Effector. You can now move the end to the chain along one of the figure's axes using the colored lines, or in the plane of the screen with the end effector.

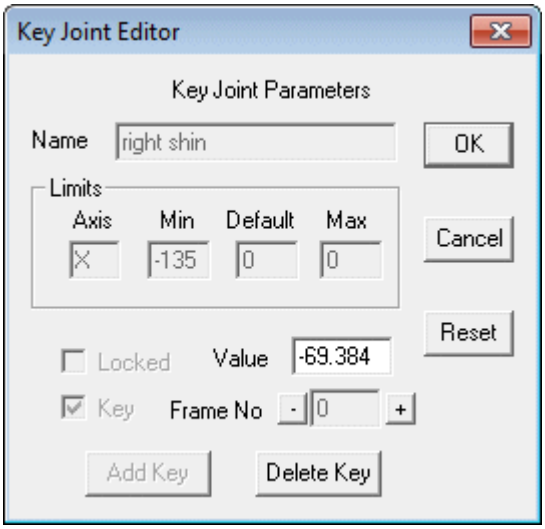
That's all there is to it! You've created a key position for a joint. To make your character move, go to another frame (click on it in the track window) and make a second key position. Click on the play button [***Cr***] to see it move.

Continue setting the positions of any joints that need changing until you are satisfied with your pose.

© 1999 - 2017 R. Steven Glanville

Editing Key Frames

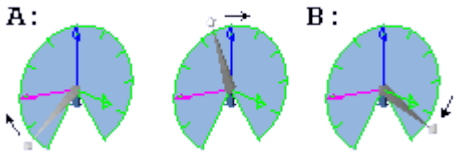
You can edit key frame values for individual joints. Double click on the joint's track in the expanded view of the track and the key frame editor dialog will appear:



The current value of this joint angle is shown in the *value* field, and the current frame next to *Frame No*. If this is a key frame for this joint you will be able to enter a new value for the key, and you can delete the key with the *Delete Key* button. If it isn't, you can make it a key with the *Add Key* button. The plus + and minus - buttons advance the current frame by one in either direction.

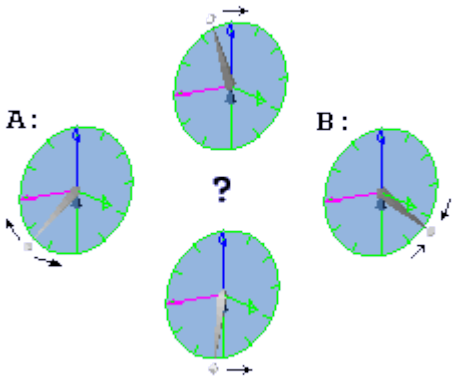
A Circle Has 720 Degrees

Sometimes rotation is not quite a simple as giving the first and last positions and letting the computer figure out what to do in between. For **constrained** joints (those with limits on how they can be rotated) like the one at the right, things are simple. Moving from position A to position B simply rotates to the right.



However for a **free** or **unconstrained** rotation this doesn't quite work. There are two directions that the joint can move: clockwise or counter clockwise.

Anim8or resolves this dilemma by ex-panding the range of angles that you can use to be from -360 degrees to +360 degrees. There are then two values that represent the same position. One-quarter turn to the right is either +90 or -270. If the difference between the angles of less than 360 degrees, the joint simply moves between the two. If it's greater than 360 degrees then the joint moves the opposite way. It's not really as confusing as it seems once you've tried it out a time or two.



© 1999 - 2017 R. Steven Glanville

Ghost Views

If you enable Ghost Views  [***G***] you can see the relative positions of other frames. Then it's easier to adjust things relative to one another. This view can sometimes get rather confusing. It can help to hide body parts and just animate the skeleton.

You can change which frames are ghosted with the Options→Ghost-Settings command. They can be limited to just key frames, or to those a fixed number of frames apart. You can also adjust the number of ghosts shown.

Making a Sequence

A Sequence can be thought of as a bunch of Key Poses played back in order. Each joint's position from one key frame is linked to the prior and following positions. Anim8or fills in joint positions for frames in-between key poses. When viewed this way, your character appears to the eye as if it is making one continuous motion. The smoothness and speed of the motion depends on how fast and in what ways the poses change. A large part of the work in animation concerns getting the motion just right.

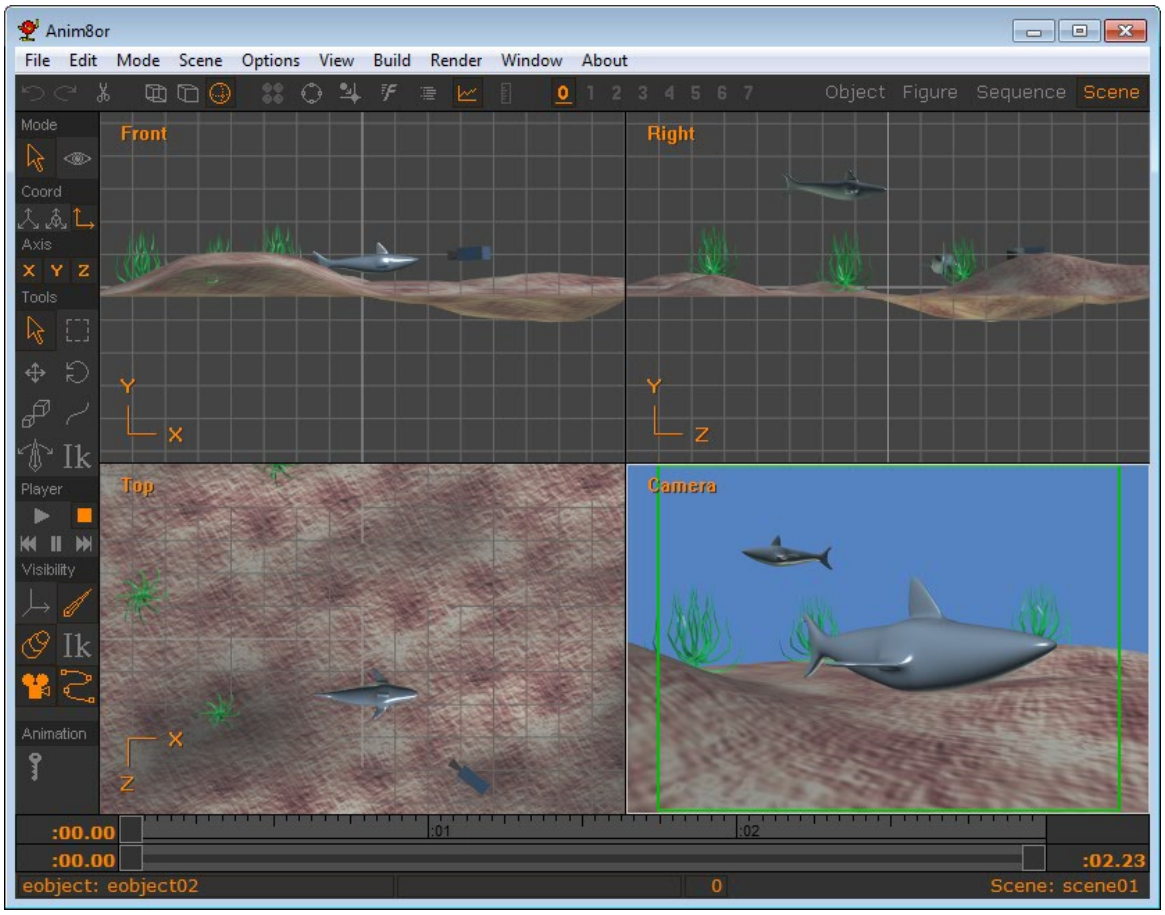
After you finish the first key pose, set the sequence to a later frame by clicking on the appropriate spot in the track, or by using the left and right arrow keys. Then make a second key pose. You don't have to set the positions of all the joints, just the ones that need it. If one part of your character is moving at a slow pace, or needs less detailed control, save yourself some work. You can always go back and add more keys if you find you need them.

© 1999 - 2017 R. Steven Glanville

Chapter 7 - Scene Editor

Scene Editor

The **Scene Editor** is where you build sets, adding objects and figures, cameras and lights, and construct your final scenes. Everything can be animated; cameras, lights, and characters. And you can save the output as a sequence of still images, or as an **.avi** movie file. You can even generate stereo views. You enter the Scene Editor from the Mode→Scene command, or by clicking on the Scene tab to the right of the top toolbar..



The basic scene editor is shown above. You can configure it to show 4 views (front, top, side, and perspective), a single view from either of these views, an orthographic view, the camera's view, or combinations of 2 or 3 views. The usual menu, toolbar, and status bar are there, and a track bar that shows the current frame.

In addition to the normal select, move, rotate, and scale buttons in the toolbar, there are several more that perform functions specific to the scene editor.

You use the **VCR** style block of buttons to control the playback of your scenes. They work in the usual fashion.

© 1999 - 2017 R. Steven Glanville

The **axis** [***X***] button makes a joint's rotational limits and axis visible when you are animating joint positions.

The **show bones** [***B***] button controls the visibility of the bones in your figures.

You can show or hide your figures' bodies with the **figure objects** [***O***] button. This can make editing joints much easier.

The **inverse kinematics** [***i***] enables editing of IK chains.

The **camera** [***C***] button controls the visibility of your camera and lights. They aren't shown in the camera's view, the one that you normally use to render scenes.

The **path** [***P***] button sets the visibility of object's paths. Any object that you select will display its path as an editable spline.

The **key** or **animate** [***K***] button enables animation. When it's pressed, the changes you make to the location, orientation, size and other aspects of objects in your scene will be animated when you play the scene.

Elements of a Scene

An **element** is a part of a scene. A new scene contains two default elements, the **world** and the **camera**. The world is the global coordinate system. The location and orientation of everything in a scene is, ultimately, relative to the world. The camera is the viewpoint for the movies you make. You cannot delete either the world or the camera.

Everything that you can add to a scene, including lights, objects, figures, and targets, are also called **elements**.

Adding Objects

You add objects to a scene by selecting the Build→Add-Object command. This brings up a dialog showing all of the objects in your project. Select the one that you want and click OK. It will be added to the center of the scene. This is not the actual object; you cannot edit it from within the scene editor. Instead it is a *link* to the object. But you can scale the view of it within the scene, and add multiple copies of it. The screen shot of the scene above has three instances of the seaweed object in it.

You will often find it more useful to select **world coordinates**  when adding objects to a scene. They are "dropped" into the scene at the center, at coordinates <0,0,0>, and are selected. Make sure that the move button  is set, and then you can use the left mouse button to drag the new object around on the ground plane from within any viewpoint. The right mouse button will raised and lower it. You can also scale  and rotate  an object's size and orientation, and animate them by specifying a series of **key frames**.

© 1999 - 2017 R. Steven Glanville

Adding Figures

You add figures in the same manner with the Build→Add-Figure command. They are dropped at the center of the scene. You must move them to their proper place. You can animate the position of figures with key frames, and you can also attach predefined sequences of motion, as described below.

The Camera

The **camera**  is the reference point for the final animation. It's fully animatable. You can change its position, field of view, direction, etc. You can also animate it for any variety of pans, tilts, or dollies. You can attach a camera to another moving object for fluid tracking shots, lock a camera's direction on a moving target, and even move right through "solid" objects. To see what the camera sees, select the View→Camera command.

You can add additional cameras to a scene with the Build→Add-Camera command. The camera's **Active** controller sets which camera is currently active. Double click on a Camera to show its properties dialog to set the Active value or use the animation track. Don't forget to enable animation with the animate key  if you want your scene to use multiple cameras!

Lights

When you first create a scene, there are two **default lights** in it. They give the scene an overall level of illumination so that you can see what you are doing. You can't move them about or change their properties. In general you should add your own **user lights** for better results. When you do the default lights are automatically deleted.

You add lights to a scene with the Build→Add-Light command. This adds a light directly overhead with default properties. Double click on the light to change its color or other attribute, and drag it around the scene to change its position or direction.

Like other kinds of objects, lights are fully animatable, including their color.

There are three kinds of lights that you can use in Anim8or:

Infinite or **directional** lights cast an even light on all objects in a scene, with all of the rays parallel, like the sun. Close and far things are illuminated with the same level of light.

Local lights emit light that radiates out in all direction from a particular point in the scene, like the bulb in a lamp. Objects between the camera and the light are backlit, while those behind the light are lit from the front. The further an object is from a local light, the less illumination it receives from that light.

Spotlights are similar to local lights, but they only illuminate objects that lie within a particular cone. Things in the center are fully illuminated, while things near to the edge of the code are gradually lit with less light.

© 1999 - 2017 R. Steven Glanville

Lights do not appear in the final scene, or in any camera view. If you want an object to appear to emit light, then you must add a light in the same location as the object. You can also toggle the visibility of lights in the other views with the show camera  button.

Targets

A target  is a dummy point of reference. You use them as points of interest for the camera to follow, or as a parent of a group of objects that you want to move. They are fully animatable, and do not appear in the final scene. You add a target with the Build→Add-Target command.

Object Properties Dialog

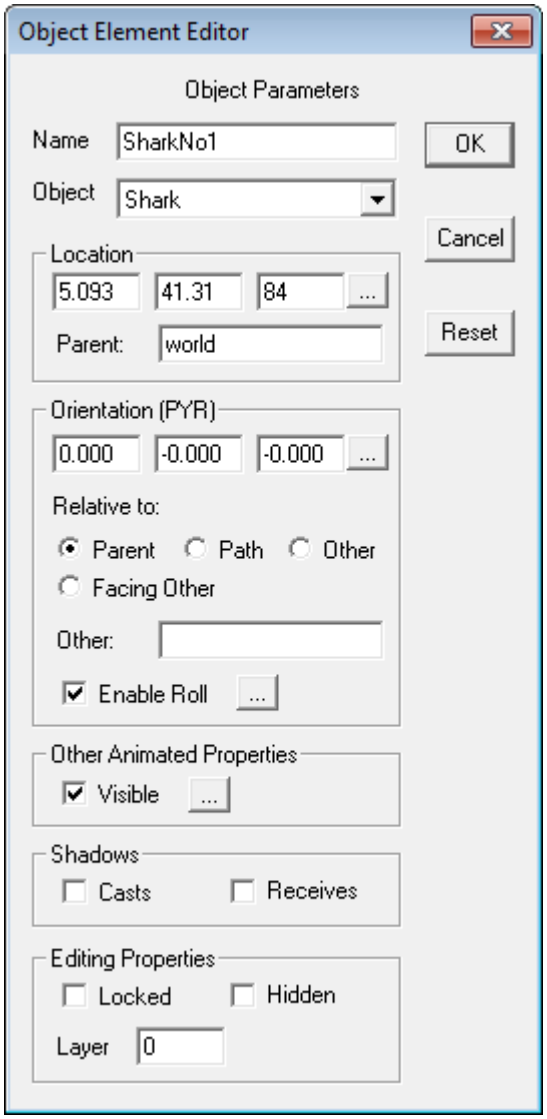
You can edit various properties of any kind of element by double clicking on it. This brings up a dialog similar to the Object Element Editor shown below.

You can animate any value that has a controller button  next to it, changing its value throughout a scene. Values with **white** backgrounds are **fixed** for the scene. You can edit them directly in this dialog. Values that are **grayed** out are **animated**. You edit their values by clicking on the adjacent controller button.

If you want to animate a fixed value, click on the button to add a controller. Controllers are explained in more detail later in this chapter.

There are several common fields to all element types:

© 1999 - 2017 R. Steven Glanville



Name

The element's name within this scene. Remember, an object element in a scene is only a *reference* to the actual object. There can be more than one occurrence of the same actual object in the same scene. If you change the original object in the Object editor then all of the references to it in your scenes will change as well.

Object

The actual element's name. You can select any object in your current project with the drop down dialog.

Location

The location of an object in <x,y,z> coordinates is shown here. This is the location relative to its parent's location and in the coordinate system of the parent. You can animate the location.

Parent

Any element can have another element for its parent. Then when the parent's position changes the element's position will change as well. With a parent, the location is a relative position. To specify a parent for an element, enter the parent's name in this box. To remove an element's parent entirely, clear the box or use the default **world** parent.

Orientation

This value shows you element's orientation in degrees of **pitch**, **yaw**, and **roll**. Pitch is a rotation around the X-axis, yaw the Y-axis, and roll the Z-axis. The values shown are a relative orientation to one of several coordinate systems. It can be either relative to its parent, to its path or direction of movement, to the orientation of another element's, or it can always facing a particular element. You can animate the rela-tive orientation.

Enable Roll

You can disable an element's roll, keeping it "upright". You normally what to keep your camera upright so by default the camera cannot roll, but everything else can. Enable Roll can be animated.

Visible

You can also animate the visibility of elements so that they appear or disappear during your scene.

Shadows

Shadows add realism to your scenes but can be expensive to compute. You can decide which elements cause shadows and which ones show shadows on them. If you want an element to cast a shadow you must enable it with the **casts** checkbox. If you want it to show shadows check the **receives** checkbox.

© 1999 - 2017 R. Steven Glanville

Editing Properties

The Editing Properties values are useful when editing your scene. You can **hide** an element by checking the "hide" box or with the Edit→Hide [***h***] command. This is useful to allow you to see what's behind something, or to simplify a scene for faster drawing. To show a hidden element, either uncheck the "hide" box, or select the Edit→Show-All [***H***] command.

You can **lock** an element by checking the "locked" box. This prevents any changes to any of its properties within the scene, such as location or orientation. This does not mean that it cannot move. If it is initially animated, it will stay that way. Locking only prevents you from accidentally changing that animation.

Note: This does not mean that the element cannot be changed in any way. For example, if it has a parent, then when the parent's location changes, so will the element's position, since its definition of location is relative to the parents.

You can also assign a **layer** to each element for use with the usual layer controls.

These values are only present for certain kinds of elements:

Scale An element's size can be animated. However you do not use a dialog to do it. Instead you simply select scaling  , enable animation  and change the size.

FOV You can set and animate the camera's **fov** or **field of view**, the angle of the width of the image the camera shows on the screen. Slowly increasing or decreasing the fov zooms the camera out for a wider view or inner for a tighter view. This value is in degrees.

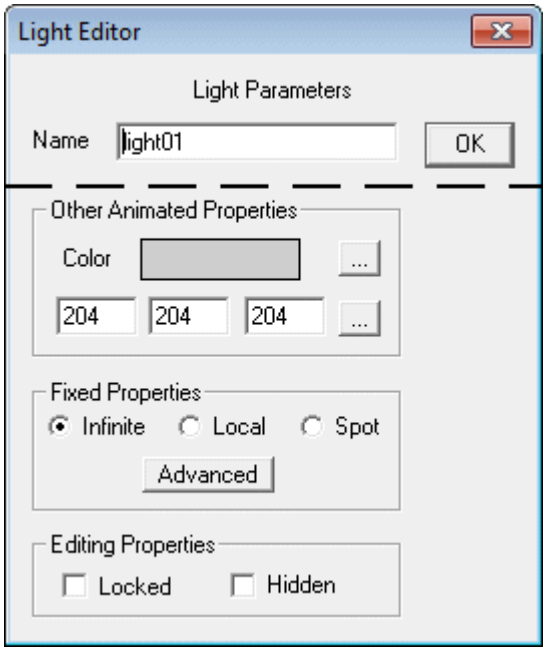
Color You can also animate the color or brightness of a light.

Light Properties Dialog

Lights have some unique settings that you can use. You access these from the Light Editor dialog by double clicking on a light.

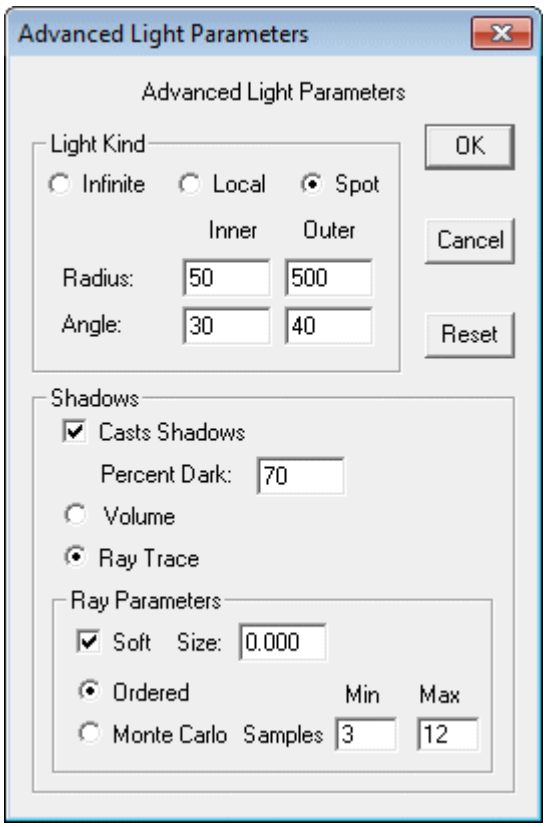
You change the color of a light by entering the numeric value directly, or by clicking on the  next to the color sample. You can animate its color with the button next to the numeric color values.

You can change your light into an infinite, local, or spot light in the Fixed Properties area. To access more advanced settings click on the  button.



 © 1999 - 2017 R. Steven Glanville

Advanced Light Properties Dialog



You can set several important lighting properties in the Advanced Light Parameters dialog. For local and spot-lights you can set limit the light's **range**. These lights have a steady brightness out to a certain distance called the **inner radius**. Beyond that distance the light's strength gradually diminishes until at the **outer radius** and beyond the light has no effect at all.

Spotlights are basically cones of light projecting from a point. You can set the angle of this cone as well as how sharply it falls off to the edge. Within the **inner angle** the spotlight has full brightness. Outside that angle the light's illumination gradually weakens until the **outer angle** where it had no further effect.

Shadow Properties of Lights

By default lights do not cast shadows. You enable shadow casting by checking a light's **casts shadows** checkbox.

Shadows in the real world are not totally dark. Some light is scattered off of nearby objects into these areas. You can illuminate the shadows in Anim8or as well with the **percent dark** parameter. This only allows diffuse light into shadows so that there are no shiny highlights, which would appear unrealistic.

There are two main kinds of shadows, **volume** and **ray traced**. You select which kind you want you light to cast in this dialog. Ray traced shadows can cast soft edges. You control how soft the shadow is with the **size** parameter. For infinite lights this is the angle that the shadow spreads from the edge of an object. For local and spotlights this is the physical size of the light. Objects closer to these lights will cast softer shadows than those further away.

You can set the minimum and maximum number of samples used to calculate soft shadows in this dialog as well.

The section on Shadows has more information on using shadows.

Shadows

Shadows are a very important part of an image. They can also consume an enormous amount of computer time. It's important that you understand how they work or you may spend a lot of unecessary time waiting for your images to render.

© 1999 - 2017 R. Steven Glanville

Shadows are only visible in rendered images made from the Scene editor. They are only cast from light that you add yourself. And they are only cast by object that you choose. To enable shadows you need to:

- Use the Scene Editor,
- Add your own light and enable the **casts shadows** in the advanced properties dialog,
- Enable **casts shadows** and **receives shadows** on the element of your scene that you want to show shadows, and
- Render an image or a movie

Volume Shadows

Volume shadows are usually much faster to render than ray traced shadows but they have some limitations. They have hard edges and you cannot use them to cast soft shadows. This tends to make your images look unrealistic.

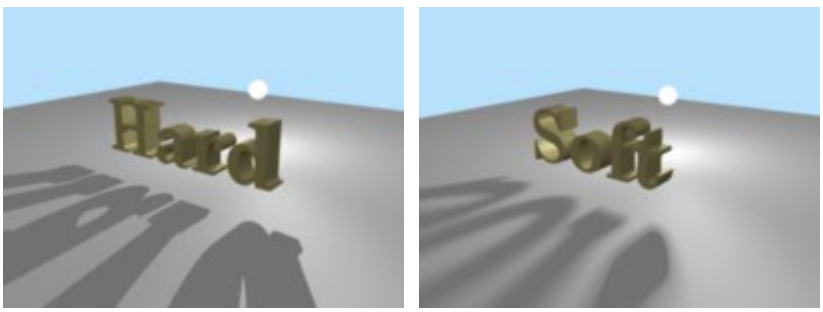
A bigger limitation is that you cannot use volume shadows to cast shadows from within an enclosed object such as a jack-o-lantern. While this is not a problem in most scenes it can cause you some unwanted "surprises" if one of your lights moves into an area enclosed by a single object. You can work around this problem by splitting the object into two or more pieces, or by using a ray-traced shadow for the offending light.

Ray Traced Shadows

Ray traced shadows are the most versatile. They can also consume a very substantial amount of time to render.

Ray traced shadows can be either **hard** or **soft**. Hard shadows have a very distinct boundary between what is inside the shadow and what is outside. Your shadow form the sun on the ground is an example of a hard shadow.

Soft shadows don't have a distinct boundary. Instead the amount of light gradually lessens until the shadow reaches its maximum darkness. The shadow cast by a florescent light is an example of a soft shadow. Such shadows are soft because the apparent size of the light casting them is large. As you can see below soft shadows can be much more realistic.



Soft shadows are costly to computer. Many individual shadow samples are used to make a soft shadow. Anim8or therefore limits soft shadows to anti-aliased images where multiple image samples are used for each pixel. High quality soft shadows often require even more samples

 © 1999 - 2017 R. Steven Glanville

than Anim8or uses for anti-aliasing. To limit the total cost of these computations, Anim8or adapts the number of samples to the image as it is being rendered. If there is nothing "interesting" in an area then only a few samples are taken. If a shadow edge is detected, however, more samples are taken to improve the image. You can set the minimum and maximum number of samples used for each light in the Advanced Light Properties dialog.

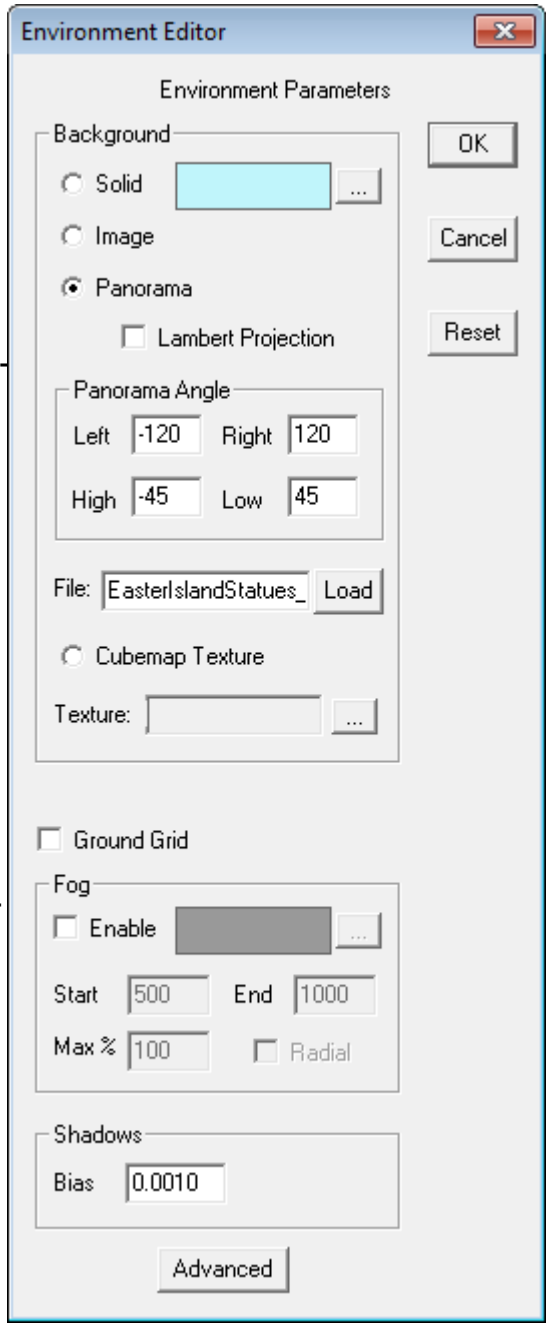
In general, you will need to set the number of samples higher for larger sized lights that cast soft shadows. You will also have to increase the number when a light is very close to an object since this will magnify its shadow's soft part.

Environment Settings

You can use the Environment Settings dialog to control several things in your scenes. You can add a **background image**. It can be either a fixed image or a **panorama** that moves with the camera pans. You can also add **fog** effects. You open this with the Settings→Environment command.

© 1999 - 2017 R. Steven Glanville

In the **background** area you can set the background for your scene several ways. By default it is a solid color. You can change the color with the ... button next to the color sample.



You can also use a fixed image for the background. To do this select the "image" button and load a background image from a file with the Load button.

You can also use an image as a **panorama**. The image is wrapped around your scene in a big arc. When the camera moves it will pan across the background. The diagram below shows how this works.

You set the area that your background fills in the Panorama Angle area. You have to be careful not to aim the camera past the edge of the panorama or the background color will show.

You can also use full 360 degree VR backgrounds. For these you need to set the angle limits to -180 to 180, and -90 to 90. There are two common projections for VR images, **linear** and **Lambert**. Lambert is used more often for scientific work because it has approximately the same solid angle for each pixel. If the background is distorted when the camera is pointing straight up or down try changing the Lambert check-box.

You can remove the default **ground grid** from your scene by un-checking the Ground Grid box.

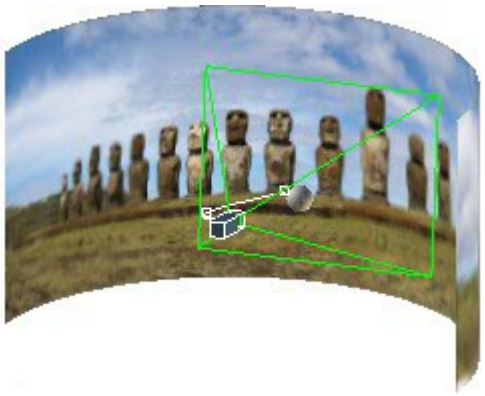
Distant **fog** can add realism to your scenes. It is useful for underwater shots as well as distant outdoor views. You enable for by checking the box at the top of the fog section of the Environment dialog. You can set the color, the distance from the camera at which fog begins to appear, and the distance beyond which it has maximum saturation. You can also limit the maximum effect it has so that things don't disappear entirely in the distance.

Shadow **bias** is an advanced technique used to hide ugly rendering artifacts that can appear when shadows are used. These artifacts are caused because the lighting computations assume that your models have smoothly curved surfaces. In reality they are made of flat faces. Shadow computations are based on these flat faces and don't always get along with the lighting computations.

To overcome this discrepancy, a shadow bias is used to push shadows slightly back away from the viewer. Then everything looks fine.

© 1999 - 2017 R. Steven Glanville

You normally shouldn't have to change the shadow bias.



Panarama Environmental Image

Note: Backgrounds only appear in the Camera view.

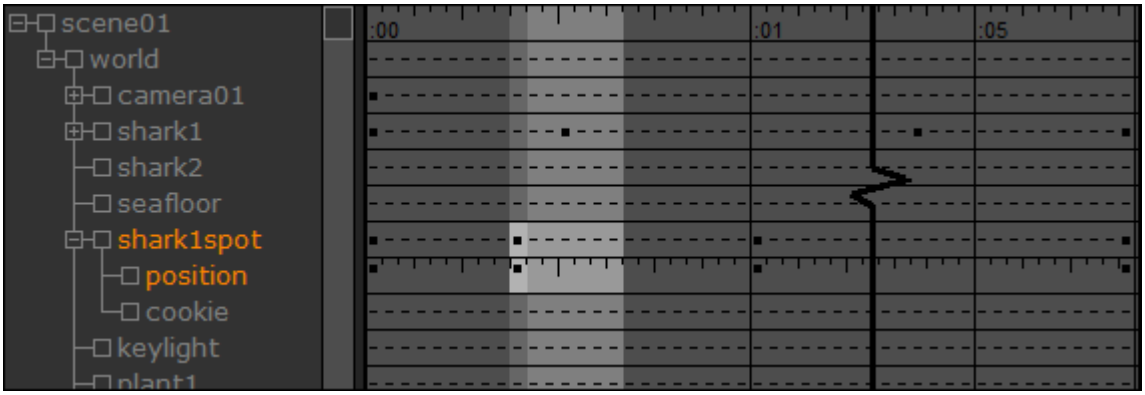
The Time Track

At the bottom of the window is the scene's **time track**. If it isn't visible on the screen, select the Options→Track-Window command:

On the left is the scene's title. It is the top of a tree description of the elements that make up your scene. If you click on the "open" box you can expand the window to show the objects contained in the scene. You can change height of the expanded track window by dragging the horizontal resize bar up or down, and the width of the name field by dragging the vertical resize bar to the right or left.

The right part of the track is a time line for the scene. If you click on it, the display will switch to the time you selected. Each tick mark represents one frame. Seconds are marked with full vertical lines. Normally there are 24 frames/second but you can change this to other frame rates with the View→Preferences dialog. The current frame is highlighted in dark gray.

You can click and drag the mouse it will select a range of frames. Use Shift-Click to select a new frame range while still keeping the same current frame.

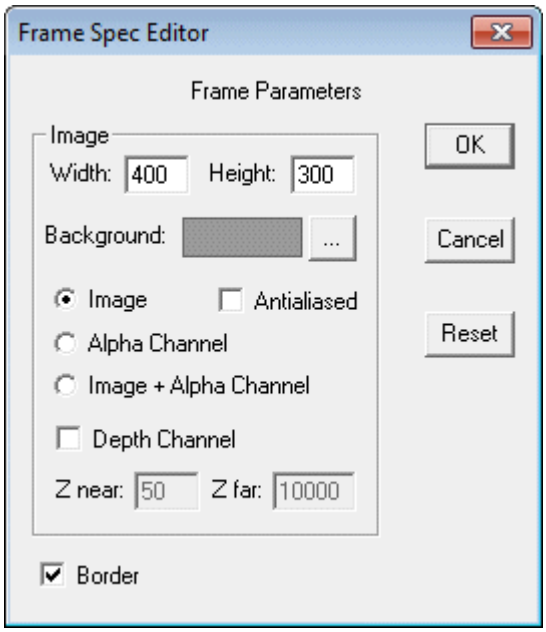


 © 1999 - 2017 R. Steven Glanville

When expanded, the left part of the track shows the objects in your scene. There is a lot of useful information here. You can see what an object's parent is, what properties have been animated, which bones can be rotated, and such. In the view above the highlighted object shark1spot's parent is the world. Its location is animated and it has a child object called cookie. If you see an "open" box ⊞ by an item then it has more information that you can display by clicking on the box.

You can select and edit items shown in the name section of the track by clicking and double clicking on them. Mouse buttons have their normal meanings with the right one adding to the selection and the middle one subtracting. Double clicking is how you can edit the numeric value of animated properties like an element scale that does not appear in a dialog.

Previewing and Saving Images



You can preview a software rendered image of any view with the File→Preview command. You can set the size and background color used to make the image, and can choose to make it anti-aliased.

You can also use the Render→Render-Image command to do this in the Scene Editor.

The Alpha Channel

When you render an image, you can isolate the elements in your scene from the background. You can save both the images of your elements and a mask showing where they are. This mask is called an **alpha channel**. It's really nothing more that a black-and-white image in the shape of your elements. It has a value of 0 where your elements are drawn, and 1 where they aren't. Transparent parts and edges in anti-aliased images will have a value in-between representing how much of the pixel your elements "own". You can use the alpha channel to combine images made in Anim8or with other images in a paint-program, as you will see below.

You make an alpha channel by selecting the "Alpha Channel" button in the preview dialog.

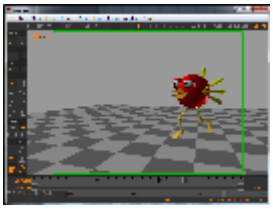
You can also make two very useful images with one rendering by selecting the "Image + Alpha Channel" button. One is an ordinary alpha channel and the other is a special element image that has been **pre-multiplied** by the alpha channel. It shows your elements normally but is black where your elements are not. Partially transparent parts of your elements are dimmed to the degree that they are transparent. You can then use these two images for a variety of special effects, such as image compositing which is described next.

© 1999 - 2017 R. Steven Glanville

Image Compositing

With **image compositing** you can insert a computer generated model into an ordinary photograph or movie, or computer generated ones. You first render an "Image + Alpha Channel" image and then use a paint program such as Adobe Photoshop to add Anim8or's output to the photo. The steps in this process are shown below. It's best to enable anti-aliasing to prevent annoying jaggies in the result.

Step 1: Render an "Image + Alpha Channel" of your model in Anim8or. The result is two images as shown here:


→

&


Camera view in Anim8or *Pre-multiplied by Alpha* *Alpha Channel*

Step 2: In a paint program, multiply the photographic image by the alpha channel image. This will mask out a hole in the photo for your model:


×

→


Photograph *Alpha Channel* *Masked Photograph*

Step 3: Still in the paint program, add the Anim8or image to previous result:


+

&


Masked Photograph *Pre-multiplied by Alpha* *Final Composite*

It's important to try and match the lighting on your model to that in the photo. Otherwise the computer generated part won't blend in very well. In this example the bobcat is lit by the sun from behind and to the right but the bird has some front lighting as well which makes it look artificial. (You would think it was a real bird otherwise, right?)

© 1999 - 2017 R. Steven Glanville

Rendering an .AVI Movie

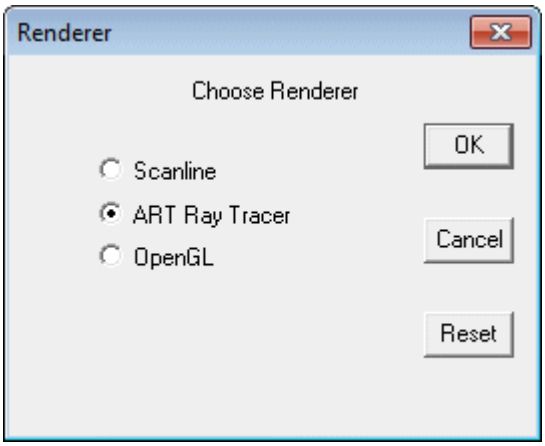
To render an **.avi** movie, display the camera's view on the screen with View→Camera, and then select Render→Render-Movie. Select AVI and any other settings you prefer, and Anim8or will render your movie.

Alternate Renderers

If your graphics card supports advanced shaders you may choose to use an alternate OpenGL renderer that uses the graphics accelerator in your computer to make images and movies. This renderer is much faster than the default Scanline renderer but it does not support all of Anim8or's options. Most notably, shadows are not supported.

Anim8or also has a built-in ray tracer renderer called ART for Anim8or Ray Tracer. See *Chapter 11: Art Ray Tracer* for more details.

You choose the renderer in the Render→Renderer dialog. If the "OpenGL" entry is not present Anim8or cannot render using your computer's graphics hardware.



 © 1999 - 2017 R. Steven Glanville

Chapter 8 - Animation

Animation

You animate the elements of your scene by setting their location, color and size etc. for a few **key frames**. The computer supplies the remaining values between these keys. You can take more control of how things animate by adding more keys. Besides their basic value some keys have additional properties that you can edit for further control.

A **segment** is a set of values for the range of frames between two successive keys. You can change how fast things move between key values by editing the properties of segments.

You can also use a **script** or **expression** to animate elements. A script is a simple program that computes the a particular value from the frame number or time in seconds, or from the properties of other elements.

Animation with Key Frames

You can animate your elements by simply moving them on the screen. First enable animation by clicking on the animate  [***K***] button. Its background turns bright blue  indicating that you are now animating. Change to a new frame by clicking on the frame in the time track. Then select an element and move it to a new location. Anim8or will add **key frames** for your element's location for these two frames and will smoothly fill in the locations for all of the frames between the keys. When you press the play button  [***cr***] your element will now move!

You will also notice that a **spline path** has been added to the scene. This is the element's path through space. You can add more points at different frames to build complex motion. You can also edit the shape of the splines by clicking on the edit spline  [***p***] button on the toolbar.

Editing objects positions is described in more detail later on. But there are a few things to note:

- When the animate button is off  any change that you make will apply to the entire scene. If you move the camera, for example, it will be moved the same amount it each frame. When the animate button is on  changes that you make will only add or change keys for the current frame. Nearby frames will be smoothly changed to reflect these changes.
- When you move or rotate an element that is a child of another element, then the change will be *relative to its parent*. The same is true when you rotate an element that is facing its path of movement. The change is relative.

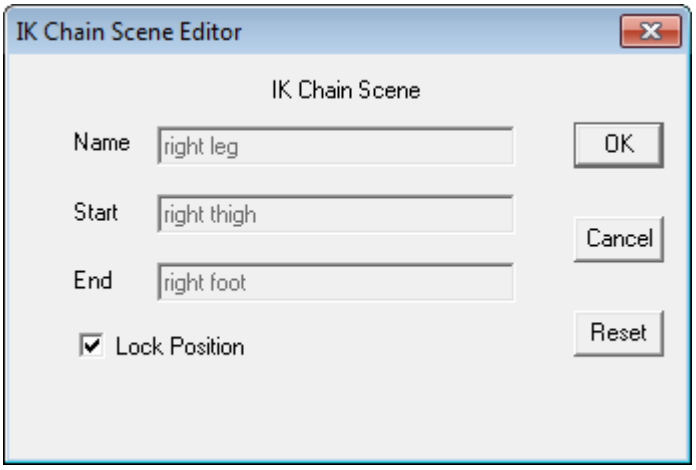
You can edit individual key's values by double clicking on the key in the time track.

Animating with Inverse Kinematics

The Inverse Kinematics tool  [***I***] allows you to set multiple keys simultaneously. As described in the Sequences chapter, you select an IK chain and drag the end effector to make your desired

© 1999 - 2017 R. Steven Glanville

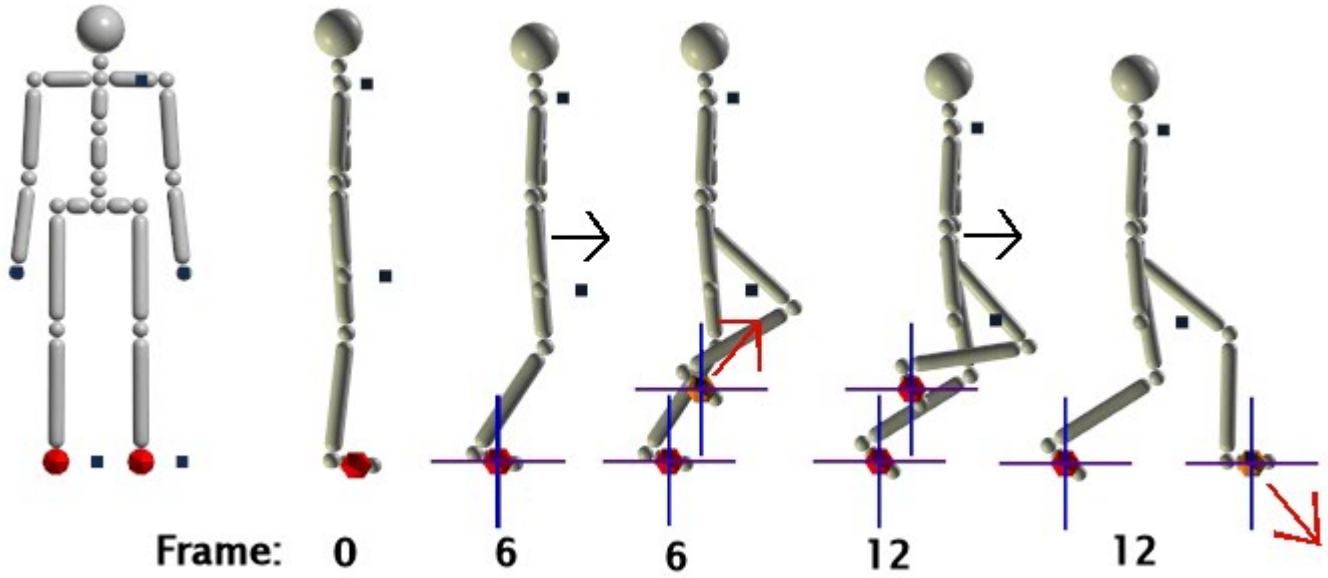
pose. However the Scene editor has a much more powerful capability. By locking the location of end effectors, when you move a figure the locked IK chains will adjust to stay as close to the locked position as possible. Double clicking on an end effector displays a dialog for locking and unlocking them.



Note: the locked property is animatable so you can change the value through-out a scene.

Inverse Kinematic Walk Cycle

Below is a simple human figure with IK chains attached to the arms, legs and back. To make the character walk, start in frame 0 and lock both leg chains. The end effectors are shown in red to indicate that they are locked.



Creating a walk cycle using IK

Change to the Right view and advance to frame 6. Move the character to the right and slightly down. Since both feet are anchored to the ground the knees will bend as shown. Now drag the right foot up and over so that it's under the middle of the character. This is now the key pose for frame 6.

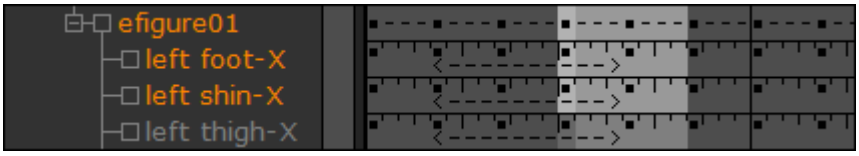
© 1999 - 2017 R. Steven Glanville

Advance to frame 12 and move the character to the right and down a bit more, then drag the right foot out in front so that it is flat on the ground. Repeat these steps for frames 18 and 24 but moving the left foot and you've created a walk cycle.

You can make adjustments to individual bones if the initial angles aren't what you want. The animation is often smoother if you do this on the same frame that you make keys using IK, but it's not required.

Animation Figures with Sequences

You can also animate a figure by adding an entire sequence of key positions from a sequence. If you make a sequence that is one cycle of a walk you can then add to a scene several times in succession to make a longer walk. Sequences appear are shown in the track bar by <----->. In the track below there is a 12-frame sequence shown for three joints.



Sequences in a scene are linked to the sequence editor. Any changes that you make to a sequence will automatically appear in your scenes. For this reason you cannot modify keys that are part of a sequence in the scene editor. If you try Anim8or will ask if you want to convert that particular reference to the sequence into individual keys and continue. Converting a sequence here will not alter the original sequence, and the link between the sequence and the scene will be lost so that further editing of the sequence will not change these keys in your scene.

Animating with Expressions

Some things can be tedious to animate an action using keys. For example, if you want a light to blink on for a short time every second you could add two keys for every second in your scene. One would set the light's color to its normal "on" value and the other would set it to black so that it would appear off. This is not only time consuming but if you make the Scene longer you must add more keys to the light or it would stop blinking before the end. If you decided that the light should be on for a longer time each flash you would need to move half of the keys.

Anim8or supports a the third way to animate elements, expressions, which makes tasks like this very simple. An expression is basically a small computer program written in Anim8or's scripting language. Each time you change frames Anim8or evaluates these programs with the time or frame number updated to their current value. The expressions then choose a value that's appropriate for that time.

This is a sample controller expression:

$color = (0.8, 0.1, 0.1)*(fract(time) < 0.2);

Used for the color of a light it would cause it to blinks red one a second. *Chapter 10 Scripts* describes the rules for writing your own expressions.

 © 1999 - 2017 R. Steven Glanville

Controllers

Controllers manage key values for animation in the scene editor. When you move, rotate, or animate any aspect of an object that uses a key, a controller is created that holds that key. Each one manages only one aspect of the scene. For example one might control the location of an object, and another the color of a light. They contain all the keys regarding that value and any related data.

If an aspect of an object is constant for the scene and not animated then it will not have a controller. The one exception is for bones. They always have a controller for each axis in each joint that can bend. This makes it easier for you to find which bones can move when you are posing a figure.

Kinds of Controllers

There are several kinds of controllers. Each uses a different kind of data, or computes in-between values in a different manner, depending on the needs of the property that is animating.

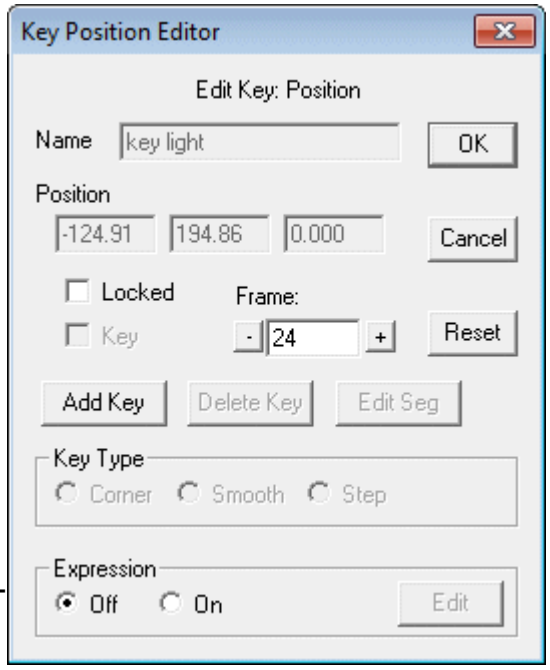
Float	This is a ordinary number like 10.5 or -150. Like most controllers, in-between values are smoothly interpreted.
Point3	Point3s return 3 numbers per frame. Each value is smoothly interpreted without regards to the others. You use these for positions and RGB colors.
Orientation	An orientation controller returns a rotation per frame. They can set the orientation of an object or can be used for a rotational direction. When you edit these values in a dialog you see the PYR (pitch, yaw, roll) equivalent to the orientation. Orientation is more complex that it may seem. Interpreting angles smoothly will not result in smooth rotation of your models. The order in which pitch, yaw and roll are applied is also important. Anim8or uses PYR because by applying roll last you can set it to 0 and you elements will stay "upright". It's often easier for you to just ignore the numeric values for orientation. Simply orient things on the screen and leave the rest to the computer.
Boolean	Boolean controllers return a "yes" or "no" value. You would use one for an object's visibility if you wanted it to suddenly disappear. There is no "maybe" value for a Boolean.

Editing Keys

You edit a key by double clicking on its key indicator ▪ in the track, or by clicking on the ... button for its controller in an element's properties dialog.

© 1999 - 2017 R. Steven Glanville

You can edit a location key's x, y and z values in a key location dialog, and add or delete keys. There are similar dialogs for the other kinds of keys. Entering a frame number lets you move to any frame in your scene, and clicking on the + and - buttons move to the next and previous frames. You can see but not change non-key frame values. The computer supplies them.



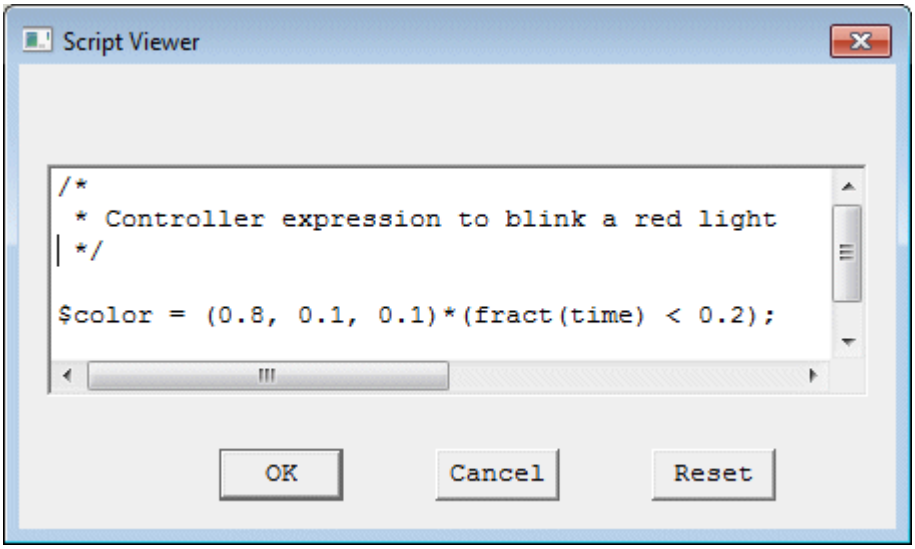
The **Edit Seg** button allows you to edit the segment next to the current frame.

There are three kinds of keys: **corner**, **smooth** and **step**. Smooth matches the direction and speed that the value changes as it passes through the key. It is what you usually want when you are animating and is the default. Corner keys can have different incoming and outgoing speeds and directions. You would use this where a bouncing ball strikes the ground. Step keys are like corner keys but allows the value to change abruptly. You would use this to make a character suddenly change positions.

To use expressions for a controler instead of keys, click the "on" button in the Expression area.

Editing Expressions

You edit a controller expression by first opening its Key Position Editor dialog, enabling expressions with the "on" radio button, and then clicking the Edit button in the expression area. This displays a window in which you can enter and edit the expression.

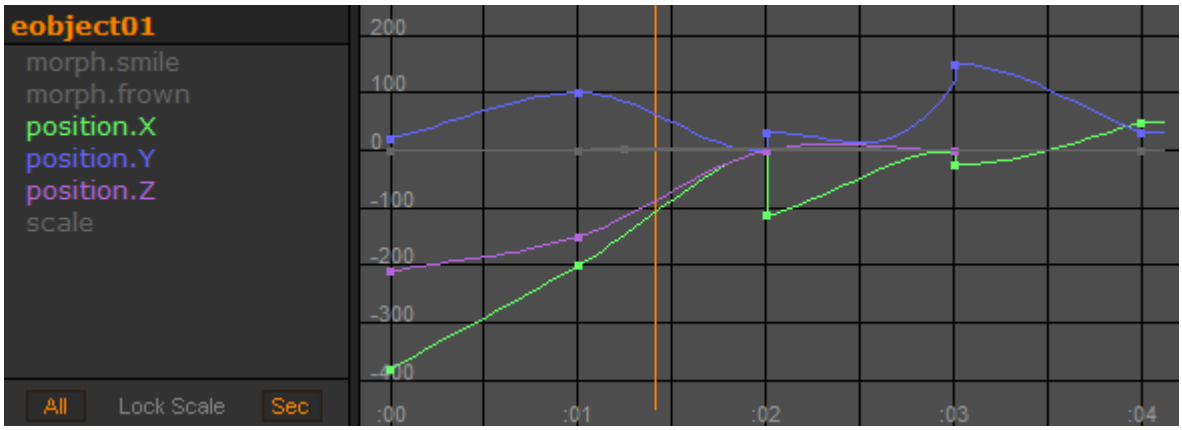


See *Chapter 10 Scripts* for more details on writing expressions.

© 1999 - 2017 R. Steven Glanville

Graph Editor

You can also edit the controller's for a scene's elements with the **Graph Editor**. It's displayed in a separate window that you open with the Options→Graph-Editor [***Ctrl-P***] command. The left panel shows a selected element's name and the names of its controllers. The right panel shows a graph of the controllers' values.



Click on the name of a controller in the left panel to select it. As usual the left mouse button selects a single value, the right button adds to the current selection, and the middle button deselects. Double clicking on a controller's name will display its key editor dialog described above.

Once selected you can edit a controller's keys in the right panel graph view. Click on a key and manipulator handles will appear. Click and drag your mouse up or down on the center handle or the lines connecting the handles to adjust a key's value. Click-drag on the end handles to change the rate that a key's value changes. The scene will be updated as you adjust the graph so you can see how much you want to adjust a value.

Handles for step keys are shown in two colors. The incoming value is in yellow and the outgoing value in white.

If the graph is too cluttered to see a particular controller you can hide deselected controller's values by clicking the All button  off. You can lock the vertical scale to its current values with the Lock Scale button  , and toggle the labels on the horizontal axis between frames and seconds with the Sec button  .

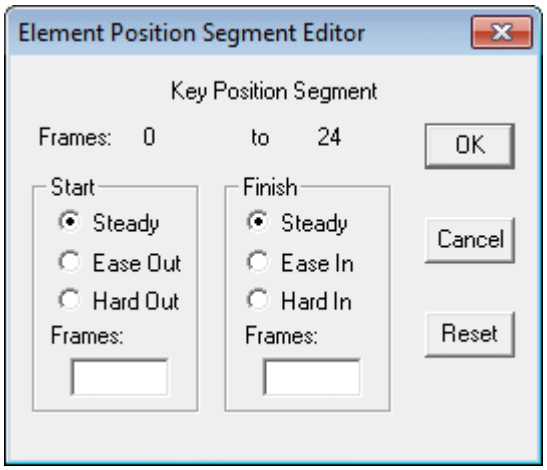
Orientation controller's may not behave as you'd first expect for several reasons. They wrap at +180 and -180 degrees so the graph can look strange as it crosses from the top to the bottom. Also changing one axis's value can affect the other axis' values. Don't worry, that's just how orientations look when show using the familiar pitch, yaw, and roll of Euler angles.

Editing Segments

Oops! Segment properties didn't quite make it into this release. These values won't do anything for you ... yet. If I get them working soon I'll make a new release but in the mean time you'll just have to be patient.

© 1999 - 2017 R. Steven Glanville

By default values vary at a consistent rate between keys. But this is not always what you want. How quickly a character reaches full speed when starting to move, or how long it takes to slow down when stopping, are important ele-ments in animation. We normally think of small, lightweight objects as accelerating rapidly to full speed, but large or heavy ones as taking a long time to build up steam. In animation terms something is said to ease out and ease in if it is slow to start and stop.



By editing the segment's properties you can change the speed a value changes in several ways. You can set the number of frames that it takes an object to change speeds, size, color, etc. at the start or end of a segment with the ease in and ease out settings. You can also select the opposite: hard out and hard in where you might want an object to accelerate at the end just before it splats into a wall to increase the visual effect.

© 1999 - 2017 R. Steven Glanville

Chapter 9 - Materials

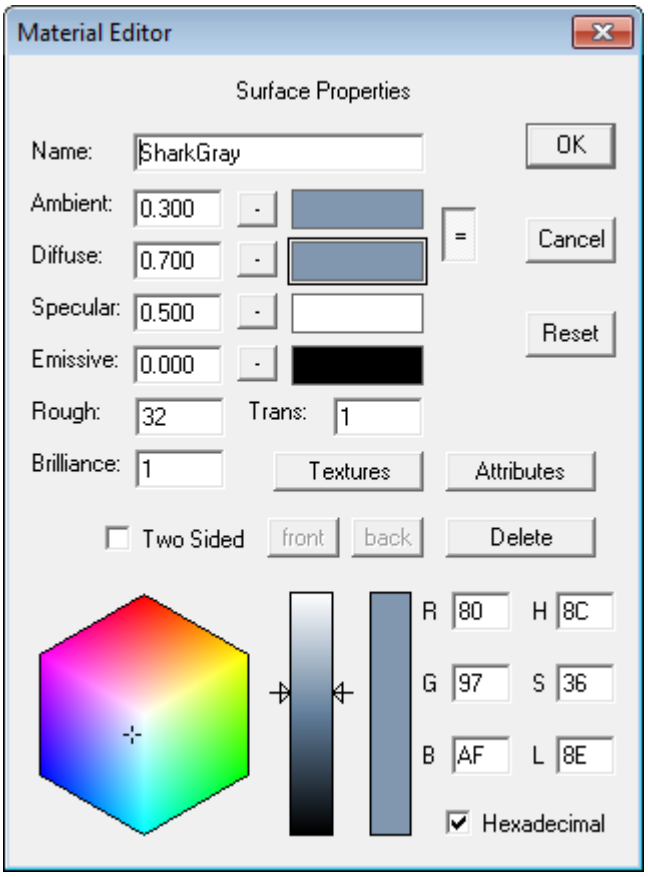
Materials

Every visible surface has a material. By changing their various properties, you set the look of your figures and objects. Materials provide a richness and detail that adds considerably to the realism of a scene. You can make them shiny, dull, transparent, and opaque. You can texture a surface with an image used as its material's color, or to lighten or darken its basic color. You can also use a texture to control the transparency.

You can't really design the look of a scene with materials alone. Its ultimate appearance also depends heavily on the lighting you choose.

Material Editor

You use the basic material editor is whenever you need to design the color, texture, transparency, and other visual properties of an object's appearance. You open the material editor either by double clicking on a material in the material toolbar, or from a specific object's editor dialog. An example is shown below.



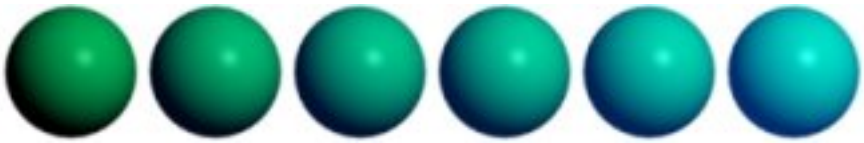
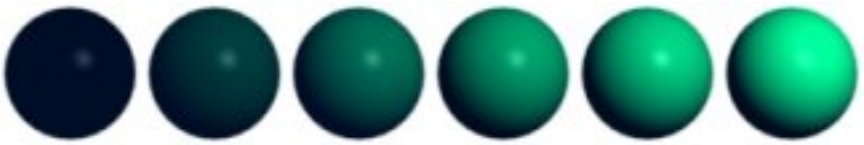
Some of the entries should be fairly obvious to most of you. Others can probably use a bit of explanation, so there's a short description of them in the following table. You can see how each property affects the look of the sample material shown in the ball below. Its parameters are all set

© 1999 - 2017 R. Steven Glanville

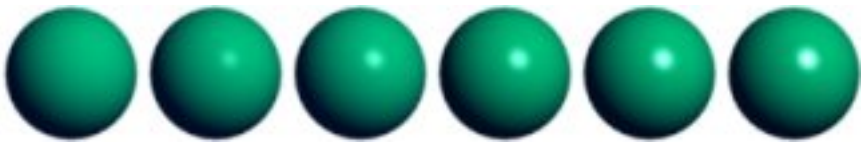
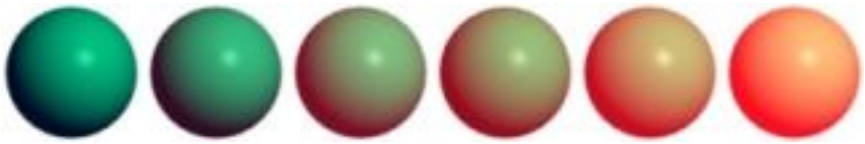
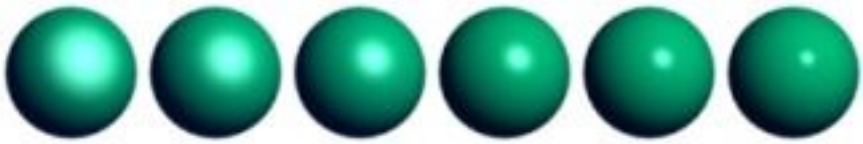
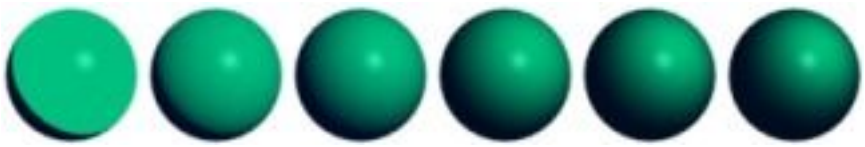
to the defaults shown above with just a single value changed for each one. Ambient and diffuse are not locked together. The base colors of the material shown to the right are:

Ambient: R=0, G=92, B=224:		
Diffuse: R=0, G=255, B=128:		
Specular: R=255, G=255, B=255:		
Emissive: R=255, G=0, B=0:		

Here is a description of the values in the Material dialog:

Name	The **name** of the material. This field can be left as the default "material00", but it is a good idea to give each material a useful name.
Ambient	The **ambient** component is the color of a material in a shadow. Normally you want it to be the same as the diffuse component. In fact this is so common that there is a button that will lock all ambient and diffuse values together so you don't have update them both to make changes. The number to the left is how much ambient component adds to the final color. The normal range for the ambient weight is from 0.0 to 1.0. The row of materials below have an ambient value ranging from 0 on the left to 1 on the right: 
Diffuse	The **diffuse** component is what you would normally call the "color" of a material. It is combined with the amount and color of light illuminating your model and added to the final color. The normal range for the diffuse weight is from 0.0 to 1.0. These samples vary the diffuse value from 0 to 1: 
Specular	The **specular** component is part of the "shininess" contribution. You normally set this color to white to reflect the light's color. For metallic surfaces you should change the color to something closer

© 1999 - 2017 R. Steven Glanville

	to the diffuse color. The normal range for the specular weight is 0.0 to 1.0. These samples vary the specular value from 0 to 1: 
Emissive	The **emissive** component represents light generated by a material. It is not affected by lighting. You use this for things like lava and eyes that glow in the dark. The normal range for the emissive weight is 0.0 to 1.0. The default value is 0.0 to prevent glowing. These samples vary the emissive value from 0 to 1: 
Rough	The **roughness** of the material. A higher value makes the surface look shinier. It is tempered by the value of the Specular component. Values range from 1.0 (not at all shiny) to 100. These samples vary the value from 2 to 64 by multiples of 2, and have the specular weight increased from 0.2 to 0.6 to show the changes better: 
Trans*	The **transparency** of the material. Actually, it's the opaqueness of the material. 1.0 is fully visible, while 0.0 is completely transparent.
Brilliance*	The **brilliance** factor. This setting changes the appearance of the diffuse component. Normal objects have a value of 1.0. If you increase it by a small amount to 1.5 or 2.0, the material takes on a sort of metallic sheen, or for brighter colors, a deeper, richer appearance. A value less that one flattens the look of the material. Here the value of brilliance changes from 0 to 2.5 in steps of 0.5: 

© 1999 - 2017 R. Steven Glanville

Textures	The **texture** button opens the general texture dialog for this material. You may use several textures on a single material, changing the diffuse color, transparency, emissive color, and more. You can also apply a bump map texture. See the Texture Dialog section for more details.
Two Sided	You can select this item, to give the material a different material for it front and back sized. The front and back buttons will show which side's properties are currently showing.
Delete	This button **deletes** a material. You will be notified if it is currently in use and given a chance to cancel the delete.
	These four color patches show the current value for the ambient, diffuse, specular, and emissive colors. The currently active one is outlined and the lower part of the window shows its numeric value. You can change the active color to a different patch by clicking on it.
- T	These buttons show you if a particular component uses a texture. If the button has a “T” on it then it uses a texture. If not then it doesn't. You can click on a button to set a particular texture.
*Note: Transparency and some more advanced texture modes may only affects rendered images. Its final effect can only be partially shown in interactive sessions. Some graphics cards can show more than others, depending on which features they support. You should render a test image to see the final look of your models.	

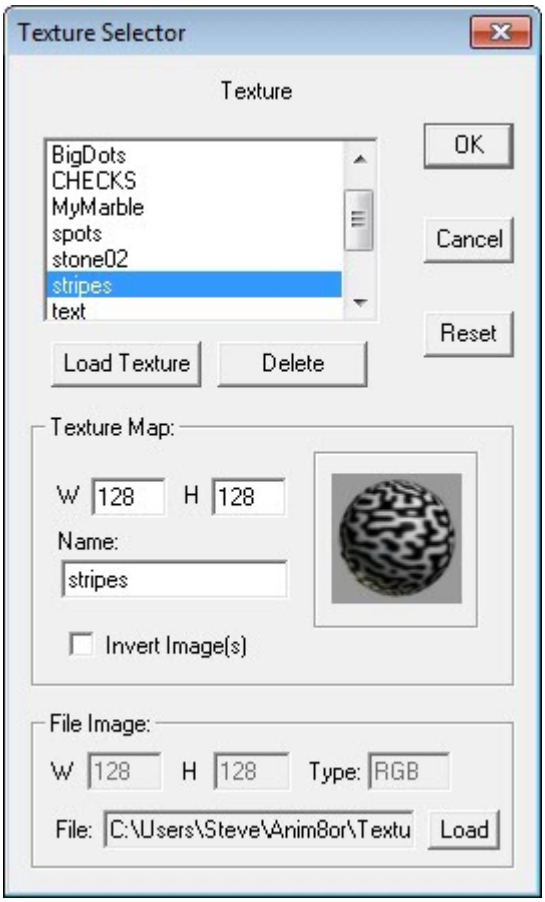
A few simple material images and their settings are shown in the table below. Notice the values for Pearl are out of the normal range. The ambient weight is more than 1, the diffuse is negative (which means lights actually darken a surface), and the specular value is way more than 1. Don't be afraid to experiment!

Sample	Material	Ambient	Diffuse	Specular	Rough	Brill.	R/G/B
	Copper	0.3	0.7	0.6	6	1.8	228/123/87
	Rubber	0.3	0.7	0.0	N/A	1	3/139/251
	Brass	0.3	0.7	0.7	8	2	228/187/34

© 1999 - 2017 R. Steven Glanville

	Glass	0.3	0.7	0.7	32	1	199/227/208
	Plastic	0.3	0.9	0.9	32	1	0/19/252
	Pearl	1.5	-0.5	2	99	1	255/138/138

Texture Selector



The texture selector allows you to load, view, and manage the textures that you use in your animations. You open this dialog from within the material editor with one of the 4 buttons:  and  . An example is shown below:

You can change the name of the texture in the Name field of the Texture Map area and can view the size of the image Anim8or uses to store it.

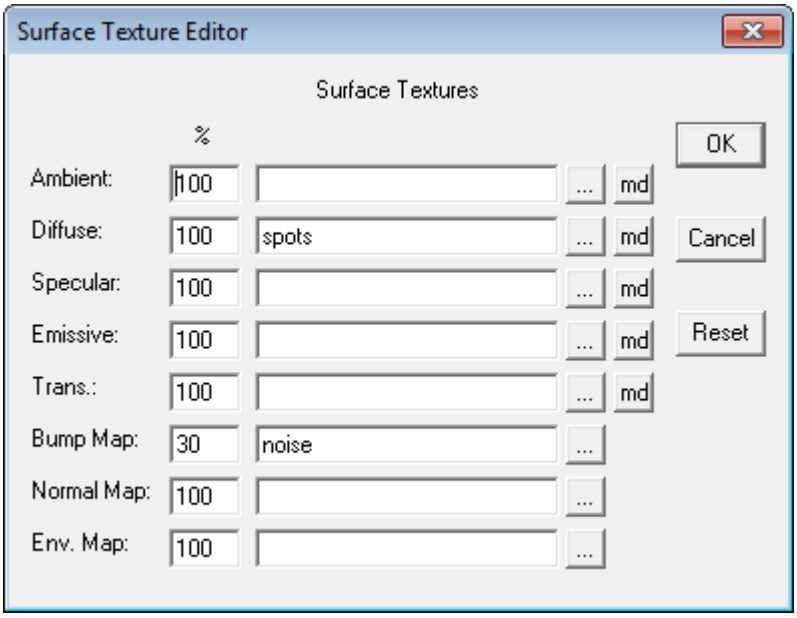
You can also invert the image in the texture by checking the **Invert Image(s)** check box.

The File Image area gives information about the file containing the texture. The Type field can be either RGB for normal images, or RGBA for .gif files with a transparent color. If your texture is type RGBA then you have more flexibility in what you can do with it.

© 1999 - 2017 R. Steven Glanville

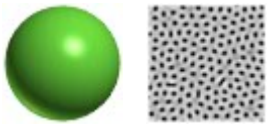
Advanced Textures

You can use textures for more than simple diffuse color maps. For example, you can map the transparency or shininess of a material to a texture. You can also use multiple textures on a material. The most basic textures can be selected directly from the material editor but you need the more advanced Surface Texture Editor for more exotic uses. You invoke this dialog using the Textures button in the material editor dialog:



The example shows a normal diffuse texture named "spots" and a bump map texture named "noise". The net effect is a bump mapped, spotty material.

The following image shows some of the surfaces that you can make with a simple green color and a single spotted texture:

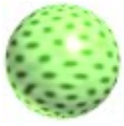


On the left are a simple green material and a black and white spotted texture. By assigning this one texture to different channels of the green material you can create a variety of diverse materials.

The diffuse channel is what you normally think of as the color of the material. If you specify a texture for this channel the material will take on its color, as shown on the upper material to the left, overriding the basic green color entirely.

You can reduce the strength to something less than 100 per cent and the two will be blended accordingly. The lower material has a diffuse texture strength of 50, which means that the final diffuse color is a 50-50 blend of the basic green diffuse value and the image in the texture.

The **specular** channel controls the strength of the high-lights. Using the spots texture here reduces the highlight where it is darker.



You can think of the **emissive** channel as glowing. Its strength is normally set to 0.0 so it has no effect. Here the emissive factor is set to 0.5 in the main material and the spots texture is assigned to the emissive channel.

© 1999 - 2017 R. Steven Glanville

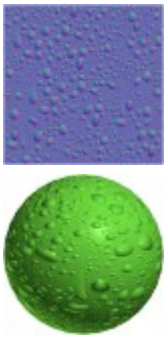
If you use the texture in the **transparent** channel, the material will have holes in it where the texture is black and be opaque where it is white or very light

The **bump map** channel changes the surface normal to appear uneven, with lighter colored areas of the texture protruding and darker ones sinking. The surface of the object is still smooth, it's just a trick of the lighting that makes it appear uneven. The sphere on the left has a 30 percent weight assigned to the texture. The one of the right has a **negative** 30 percent weight which inverts the direction of the "bumps".

Here is an example of what you can do by assigning textures to multiple channels. In this case the same texture is assigned to the diffuse and the bump map channels, though you can just as easily use different textures. There is one more trick: the texture **blend mode** is set to darken which makes the diffuse channel the product of the texture and the basic green color. See the section on Texture Modes later in this chapter for more details.

Here are some more examples of using a single texture for both the diffuse color and bump map. It doesn't always work, especially when there are strong shadows in the image, but it's easy to do:

Normal Maps



A **normal map** is similar to a bump map except that the colors in the texture represent a rotation of the surface normal instead of the height of the surface. Here is an example normal map representing drops of water of a surface, and the same green material with it assigned to the normal map channel.

There are several ways to define normal maps. The most common uses tangent space normal. The red channel represents a rotation in the U or binormal direction relative to the surface, with 0.5 or 127 representing no rotation. Smaller values rotate the normal in the negative U direction, larger ones in the positive direction. The green channel does the same for the V or tangent direction.

© 1999 - 2017 R. Steven Glanville

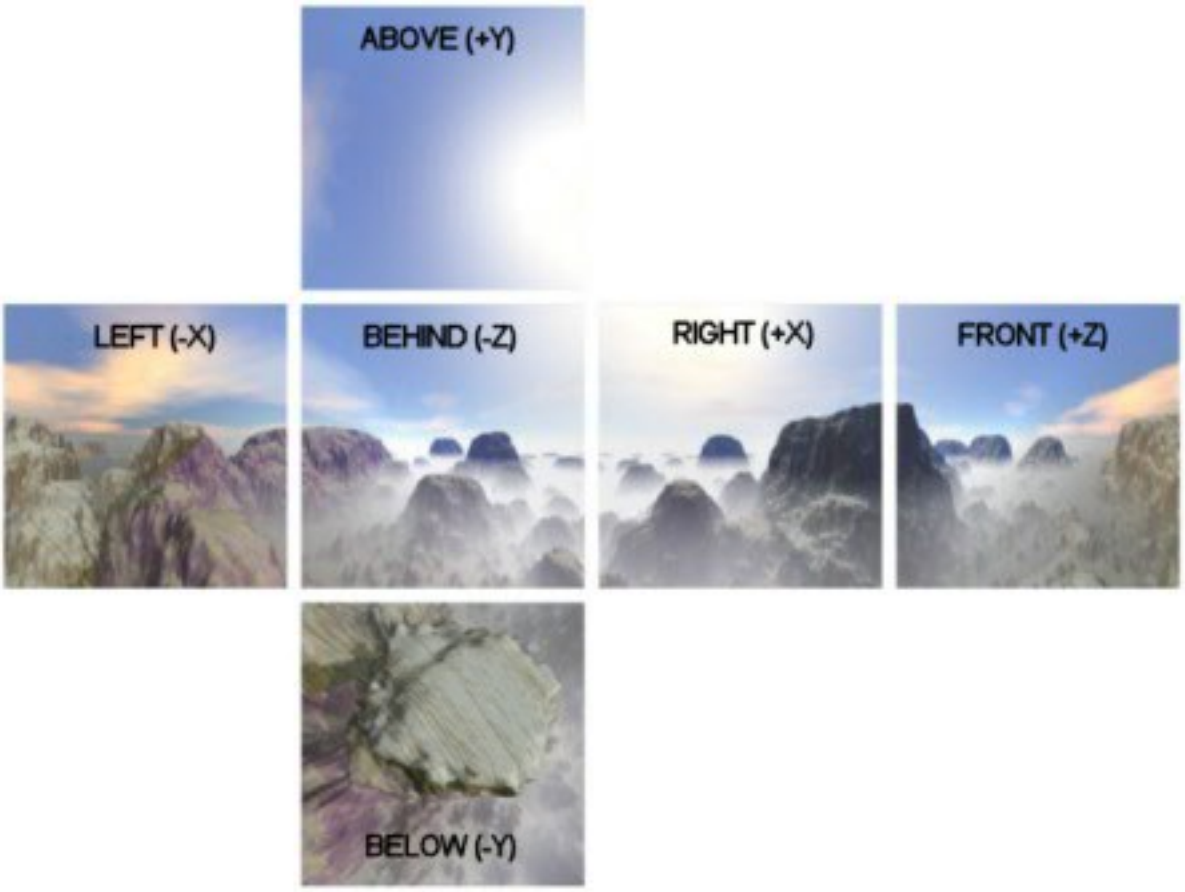
Note: prior to versions 2.51 Blender used reverse U and V rotations, which make the surface appear indented where it should be raised.

Environment Maps

An **Environment Map** texture is a representation of the surrounding world. You use them to simulate shiny materials like chrome, and to show general reflections on an objects like the square shape of a window reflecting on a cue ball. Environment maps are not actual reflections of the rest of a scene. Instead they are intended to give the *illusion* that they are reflecting the scene.

For example, the image to the left uses an environment map of a view of mountains. The sphere appears to be made of chrome and reflects a view of the mountains and a chrome sphere would. It does not reflect either the checkerboard ground or the nearby 3D letters. These features are not present in the environment map so they aren't visible in the apparent "reflection".

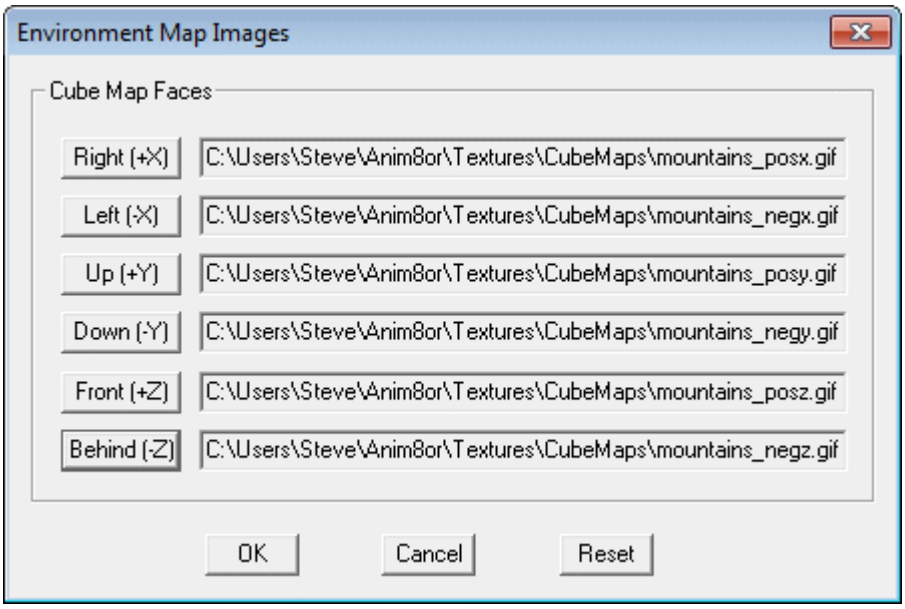
The best way for you to create an environment map material is to use a **cube map** texture. These textures are composed of six different images that form the sides of a cube. Each one represents the view of the environment around your scene in a different direction. Think of it as if you are inside of a large cube. The six images used in the example above are shown below.



 © 1999 - 2017 R. Steven Glanville

The directions are relative to the an object within the scene when the viewer looking into the **front** modeling view. So **behind** (**-Z**) means behind the object, in the -Z direction.

You add cube map textures by clicking on the ... button on the Env. Map row of the Surface Textures dialog. This brings up the Texture Selector dialog in Environment Map mode. There you click on the usual Load Texture Load Texture button but instead of prompting you for a single file you are asked for six files:



There are a few things that you need to know about cube maps:

- The standard cube map format has the images flipped top-to-bottom. So if you find a set of cube map images that appear upside down don't worry. They will work just fine. However you can use a right side up set as well. Just check the "Invert Image(s)" box in the Texture Selector dialog and they will be flipped for you.
- Cube map images should be square. Rectangular images will not work properly, particularly in the working views.
- All six faces should be the same size. If any of the faces are a different size the texture won't draw properly.
- Cube maps require a lot of memory. Use smaller images if you can get away with it.
- As with all textures, you will get the best results if your images are a power of two in width and height. So make your maps 256 by 256, 128 by 128, etc. if you can.

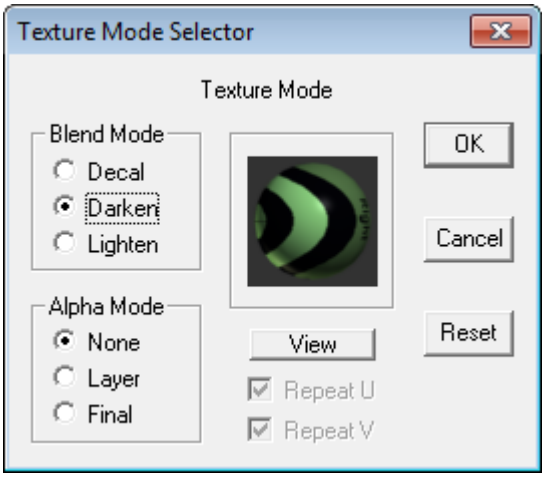
Environment Map textures are always added to any the base material color. Thus you would normally set the color of a simple Chrome material to black if you are using an environment map. Of course you can experiment with other settings too, to see what they will make. Here are some other environment map textures:

 © 1999 - 2017 R. Steven Glanville

You can also use a single image as an environment map. In this form the image is used as a latitude-longitude map of the environment. You can often paint a simple image to use for such a map. The one at the right makes a reasonable chrome reflection for an illustration.

Texture Mode

You can blend textures with the base color of a material in several ways. Textures with an alpha channel are even more versatile. You set the way textures behave in the texture mode dialog, which is reached with the  mode button in the material editor.



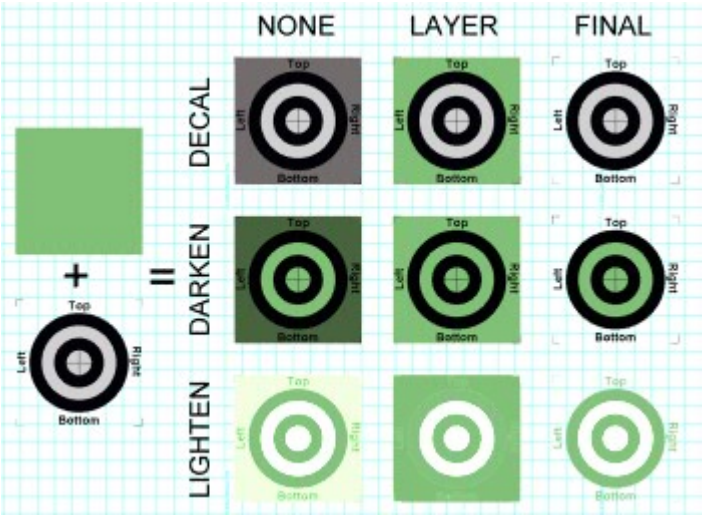
You use the **blend mode** to control how the color of the texture changes the base material color. **Decal** simply replaces the color with the texture. **Darken** multiplies the value of each color component which tends to darken the image. **Lighten** adds the two values which can only make the image brighter

If the texture has an **alpha** channel (such as a transparent **.gif** has or an RGBA **.png** file) they you can use it in three different ways by setting the **alpha mode**. **None** simply ignores the alpha. **Layer** uses the alpha channel to blend between the texture's color and the material's

 © 1999 - 2017 R. Steven Glanville

color. **Final** uses the alpha channel as the material's final transparency value.

Below is an example of what can be done with one texture in the diffuse channel and a surface color of green. The background in the target texture, where the blue background grid is visible, has an alpha value of 0 so it is transparent.



Some aspects of a texture mode are only meaningful for specific texture uses. For example bumpmaps don't use an alpha channel so the Alpha Mode set-tings don't have any effect on them.

 © 1999 - 2017 R. Steven Glanville

Chapter 10 - Scripts

Scripts

You can extend Anim8or's basic functionality and automate repeated tasks with **scripts** written in a programming language built into Anim8or. It is designed to make animation easier with features unique to Anim8or's operation.

There are several places that you can use scripts and they are all written in the same basic way. While they are all written in the same basic way, certain aspects are only al-lowed for specific kinds of scripts. For example you can't refer to a Figure in the Object Editor, and the time() function is not meaningful outside of a Scene.

Scripts may be used in the following places:

- Object text file export plug-ins,
- Parametric Shape plug-ins,
- General Object construction, editing and rendering in the Object editor,
- Controller expressions in a Scene.

There are other useful places for scripts that will be added in future releases.

This chapter provides an overview of scripts and the basics of using them in Anim8or. A more detailed language definition is provided in the *Anim8or Scripting Language Manual*.

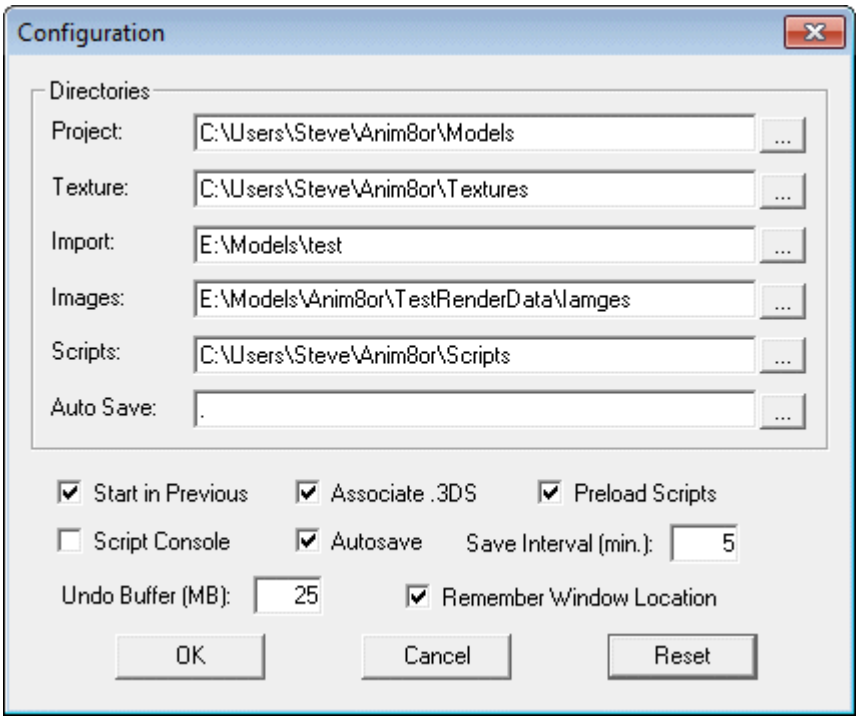
Creating a Script

A script is a text file that has either a **.txt** or **.a8s** extension. You may use any text editor to create and modify them, or you may create the within Anim8or. Normally a script will have a .txt extension. If it has a .a8s extension and you place it in the "Scripts" directory set in the File→Configure dialog Anim8or will attempt to preload it when it begins execution. If the script is appropriately defined it will become a plug-in. Alternately, it can be added as a command in the appropriate Scripts menu for easy execution when you are modeling or animating.

The Script Directory

You may configure Anim8or to preload scripts from a directory called the scripts directory when it begins execution. This is where you put any plug-ins that you want to use and common modeling scripts. You set the directory in the File→Configure dialog and need to make sure that the Preload Scripts box is checked.

© 1999 - 2017 R. Steven Glanville



When preload is enabled Anim8or will read any files in this directory that have the extension **.a8s** and parse them as scripts. If a file has no errors it will be added to the Anim8or interface. Parametric shape plug-ins will go in the left hand toolbar. Object export scripts will be added as additional choices in the Object→Export dialog and Object command scripts will be added to the Scripts menu. All other files with the .a8s extension will be parsed for correctness but not saved in the computer's memory. You can run them with the Scripts→Run-Script-File command at any time.

Installing a Script

All you need to do to install a script is to copy it into the scripts directory that you have chosen. Anim8or will load is when it starts running. It will also list the scripts read in the command window and any errors that it finds.

You may also store other scripts in this directory for convenient access but not have them preloaded by changing the file extension to .txt. You can edit and run then from with in Anim8or with the Scripts→Edit-Script-File and Scripts→Run-Script-File commands. You can also load modeling scripts with the Scripts→Load-Script-File command. It's not possible to load additional plug-ins while Anim8or is running.

Plug-in Scripts

Plug-in scripts work exactly like commands that are built into Anim8or. The only difference is that they are written in the Anim8or scripting language and can be added or removed as you wish.

A plug-in script must be written to perform a specific task. Early in the script it will have a directive declaring what kind of plug-in it is and other related details. This is how Anim8or knows where to "plug it in". Additionally each kind has specific rules that it must follow or it would not work correctly.

 © 1999 - 2017 R. Steven Glanville

Parametric Plug-in Scripts

Parametric plug-in scripts allow you to add new basic modeling shapes to the Object editor. They are similar to the built-in spheres and cylinders in that you define their shape by a few numeric values. Anim8or executes a parametric mesh's script with updated values each time that it needs to update the geometry.

It's important to understand that when you save an .an8 project that uses a parametric shape plug-in you will need to have that plug-in installed when to properly view your model at a later time. You can load and edit a project with a missing shape plug-in but you won't be able to see or alter those shapes during that session. This is not the case for modeling or export plug-ins. Once you run of these they will make a change to your project or write an output file just like you used other similar commands.

Export Plug-in Scripts

Plug-in scripts that export objects appear in the Object→Export command's "Save as type:" drop-down menu along with the built-in exporters. They are indicated by the addition of ":plug-in" after the description.

Running a Modeling Script

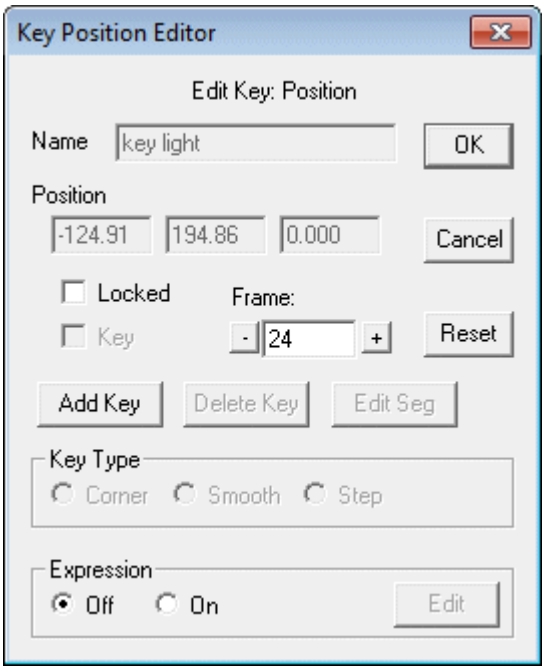
A modeling script is simply a script that is not a plug-in. They contain commands like you would use to build a model or make a material. When you run a modeling script it will change your project it some way and exit so that you can continue working. Normally this will happen very fast but it is possible to write a script that will take a long time to complete. Once you've become familiar with a particular script you will know what to expect from it.

You run modeling scripts from the Scripts menu. Those with .as8 extensions from the scripts directory will be listed in the menu. You can run other scripts from anywhere on your computer with the Scripts→Run-Script-File command.

Controller Scripts

You can also use scripts instead of key-frame splines in controllers. To change a particular controller to use them select the "On" button in the Expression area in the key frame dialog. Anim8or will supply a very simple script to set the value of the controller, normally to 0. You may edit the script with the Edit button.

 © 1999 - 2017 R. Steven Glanville



Script Errors

If Anim8or detects an error while running a script it will print a message to the command window. If you are able to write scripts this may help you find and fix the problem. If you are running a plug-in then you should contact the author for a fix.

Writing Scripts

Script writing is simply writing a program in the Anim8or Scripting Language, ASL. You can learn to write simple scripts fairly easily, particularly controller expressions. Complex scripts can require advanced programming skills. This manual does not explain how to write scripts. However if you know how to program in C or Java you should have no problem writing scripts.

A simple summary of the major aspects of Anim8or's scripting language is described in the following sections. The full syntax for scripts is described in the *Anim8or Scripting Language Manual*.

Data Types

There are quite a few predefined types in scripts. You will recognize some of them but many are specific to Anim8or. Some common types are:

int - 32 bit signed integer
float - single precision floating point number
point3 - a vector of 3 floats
object - an Object
figure - a Figure
material - a Material
shape - a component of an Object such as a Mesh or Sphere

© 1999 - 2017 R. Steven Glanville

Variables

Scripts use typed variables. You must declare a variable and give it a type before you can use it. Built-in variables and functions use normal C rules for their names. That is they begin with a letter or an underscore "_" which can be followed by letters, underscores and digits. Examples are:

scale cos AddPoint face2 KEY_FRAME _exit

User variables begin with a $ character followed by a C name:

$counter $NewIndex $face2 $_selected_index

Variable declarationas are like those in C. Variables can be single values or arrays. Unlike C, you cannot initialize a variable in a definition however.

```
float $size, $width;
int $i, $counter;
shape $mySHape;
int $index[10];  /* array initially of 10 ints. */
```

Arrays are much more flexible than those found in most languages. You can change the number of elements they hold in a running script, add and delete elements, and do other operations. There is more about arrays later on in this chapter and in the reference manual.

Expressions

ASL supports many of the operators found in C.

Function Calls

ASL has a number of built in member and non-member functions. These are described elsewhere in this document. User functions must be defined before they are called and recursion is not supported.

Unary Operators

- negation: int, float, point2, point3, quaternion
! not: int, float
~ bitwise not: int
++ -- prefix increment, decrement: int, float; note: returns l-value

© 1999 - 2017 R. Steven Glanville

Binary Operators

+ addition: int, float, point2, point3, quaternion
- subtraction: int, float, point2, point3, quaternion
+ concatenation: string
* multiplication: int, float, point2/3*float, float*point2/3,
quaternion*float, float*quaternion
/ division: int, float
% mod: int
<< >> shift: int
< == <= comparisons: int, float, string
> != >= returns int with a value of 1 or 0
&& logical AND: int, float, returns int 1 if both operands are non-zero
|| logical OR: int, float, returns int 1 if either operand is non-zero

Statements

Scripts support several kinds of run time statements. Many are similar to C but not all. Statements can be nested to any depth. There is no goto.

The basic statement is an **assignment**. It is like C's except you cannot cascade multiple assignments in the same statement. The value of the expression is computed and saved in the variable at the left:

<variable> = <expression>;

Compound statements group several statements together into a single statement. When encountered each statement within a compound statement is executed in order. They are useful when you need to use more than one statement as part of a structured statement.

```
{
  <statement>
  <statement>
  ...
  <statement>
}
```

An **if statement** allows you to conditionally execute one or more commands. There are two forms one with an else and one without one. The expression is first evaluated. If it is not equal to zero then the <then-statement> is executed. If there is an <else-statement> it is skipped. If the value of the expression is zero then the <else-statement> is executed if one is present. The full statement is:

```
if ( <expression> )
  <then-statement>
else
  <else-statement>
```

and the form without an <else-statement> is:

© 1999 - 2017 R. Steven Glanville

```
if ( <expression> )
  <then-statement>
```

A **while statement** repeatedly evaluates an expression and executes a statement as long as the value of the expression is non-zero:

```
while ( <expression> )
  <statement>
```

A **for statement** executes its sub-statement multiple times. For statements are not like those in C. In a script the increment and limit used with the control variable are computed prior to executing the sub-statement and saved. These values are used each time the body of the for statement has been executed to update the control variable and to test to see if the loop should be executed again. The syntax of a for statement is different from C to help remind you that it does not follow C's rules. There are two forms. The one without the <step-expr> uses a step value of 1:

```
for <variable> = <init-expr> to <limit-expr> step <step-expr> do
  <statement>
for <variable> = <init-expr> to <limit-expr> do
  <statement>
```

The <variable> must be declared prior to the for statement and must be an int or a float. Initially the <init-expr> is evaluated and assigned to the <variable>. Then the <limit-expr> and <step-expr> are evaluated. If there is no <step-expr> a value of 1 is used. The value of <init-expr> is compared with that of <limit-expr>. If it is greater and the value of <step-expr> is greater than zero, or if it is less than and the value of <step-expr> is negative, or if the value of <step-expr> is zero then the execution of the for statement is complete. Otherwise the <statement> is executed, the saved value of <step-expr> is added to <variable> and they are tested in the same manner to see if the <statement> should be executed again.

User Functions

User defined functions are similar to those in C, with two main exceptions. Functions must return a value, and recursion is not supported. The syntax of a function is:

```
<type> <ident> ( <argument-list> )
{
 <statements>
 return <expression>;
}
```

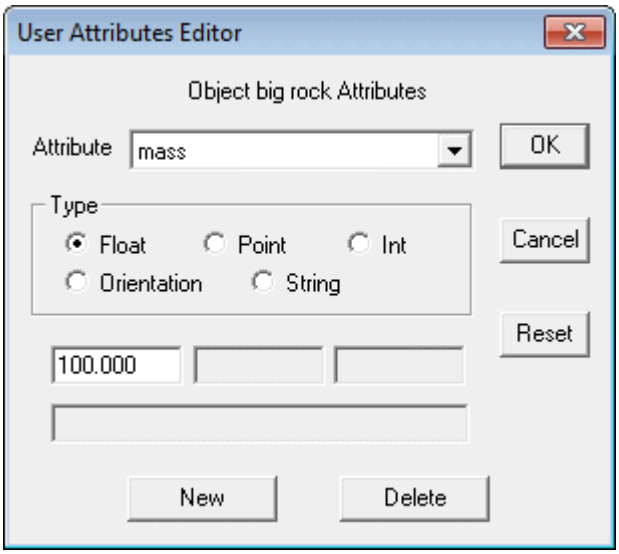
You can have multiple return statements. You should have one on all exits from the function or the result will be undefined.

User Attributes

A **user attribute** is a custom property that you can add to objects, meshes,materials, etc. An attribute associates a name with an int, float, point3, quarternion or string value. For example you

 © 1999 - 2017 R. Steven Glanville

can define a float value called mass and set it to 1.5. They are animatable and useful as an auxillary source of data for scripts. The dialog for defining attributes is shown below:



The attribute "mass" is a float with the value of 100. Anim8or doesn't understand these values but you can query and change their value in controller scripts to control how your characters behave with the GetAttributeFloat() and similar functions. See the *Anim8or Scripting Language Manual* for details.

 © 1999 - 2017 R. Steven Glanville

Chapter 11 - ART Ray Tracer

ART Ray Tracer

Ray Tracing is a global rendering technique for making relatively realistic images from 3D models. It traces the path that a light ray might take through the scene to discover what objects, lights, etc. affect the final color of a pixel. It is called a global technique because any part of the scene may influence to final value of any pixel via reflected rays of light. Ray tracing borrows from optics to model reflections, refractions and more.

Anim8or has an integrated ray tracer. You can use it to render movies and still images just like with the original scanline renderer and OpenGL shader renderer. All you need to do is select it from the Render→Renderer command dialog.

Anim8or's ray tracer can render all of its normal materials, of course. But it also can show reflections off of shiny objects and diffraction through transparent ones. The next sections will explain how to add these effects to your images.

Art Materials

You enable the additional capabilities of ART materials by adding certain **attributes** to normal materials. Attributes allow you to associate arbitrary values with a material. The ART renderer checks them when initializing a scene and chooses alternate rendering code when the right ones are present. You add attributes to a material by clicking on the "Attributes" button 

in the basic Material Editor. This section lists the main categories of ART materials.

The attribute class is the starting point for accessing ART materials. class is a string attribute which defaults to normal Anim8or materials when not present. *Note that case is important in attribute names.* Class and CLASS are names for different attributes and do not influence ART rendering. In addition to the normal parameters of Anim8or materials, individual classes may use additional attributes for more flexibility.

The main classes are anim8or (the default), glossyreflector, transparent and dielectric. Each is described in a following section.

Class anim8or

Class anim8or has an additional string attribute, specular. It determines the look of the specular reflection. There are 3 recognized values:

- phong - the default value, phong is simple normal Anim8or phong shading for highlights.

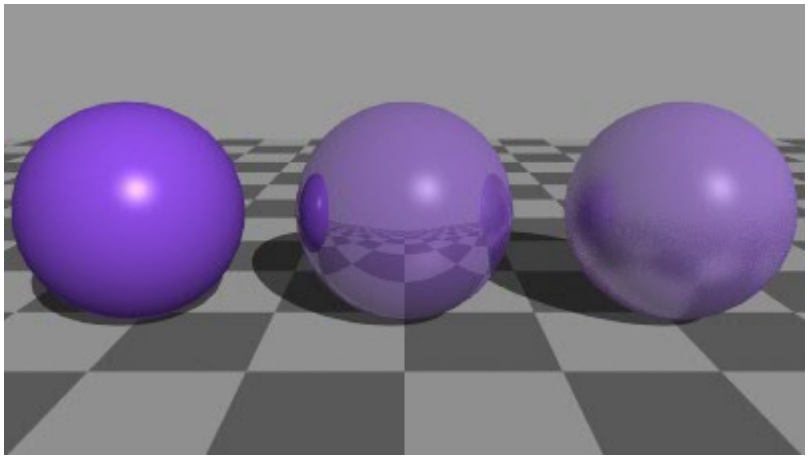
- reflection - a mirror like perfect reflection is added to a material. Rays are reflected off of a surface to determine what is visible in that direction. The result is multiplied by the materials specular weight, Ks, the material's normal color - including the original phong highlight - is weighted by (1 - Ks) and the two values are added for the final color. Thus a value of Ks

© 1999 - 2017 R. Steven Glanville

= 1.0 would be a perfect mirror with no base material properties visible, and Ks = 0.1 would be a normal material look with a slight reflection of the scene.

- glossy - an imperfect reflection is added to the material. This is like a blurred mirror or defocused camera image. The further from a surface, the less distinct the reflection. The overall amount of blur is determined by the Phong Roughness. The default value of 32 is very rough. Higher values like 1000 make a more perfect reflection while 100000 is almost the same as a perfect reflection. Glossy images need anti aliasing turned on or they will look very speckled. Extremely glossy images will need more that the default AA samples per pixel value of 16. The image below uses the default, 32, as the default samples per pixel of 16 and is rather grainy.

Below is an example of the three settings for the specular class. From the left they are phong, reflection and glossy. Ks is 0.5 and Roughness is the default of 32.



Class glossyreflector

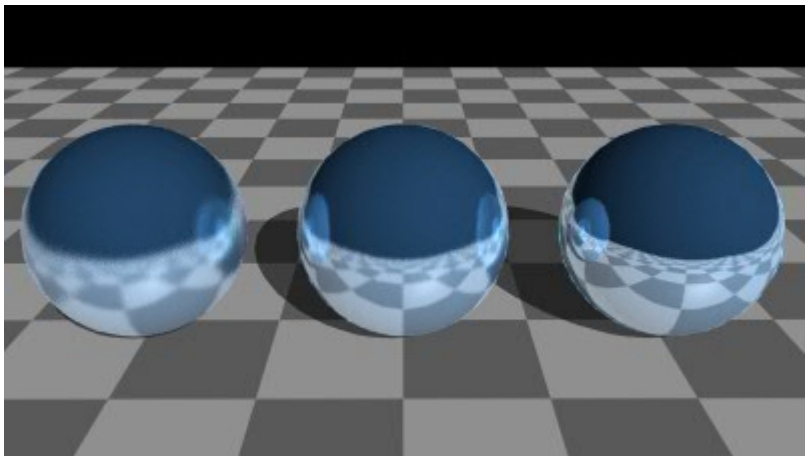
This glossyreflector class is a simpler class that is similar to the anim8or class with the specular attribute of glossy. While the anim8or class is a little less accurate from a physical sense, glossyreflector attempts to follow the rules of physics a little more closely.

The glossyreflector class uses the Phong roughness value for the degree of glossiness. glossyreflector has one additional attribute:

- Kr - constant of reflection. This is similar to the specular constant in normal Anim8or material's, Ks, but controls what percent of the light is reflected. It is set to the value of Ks if there is no explicit Kr.

Below are three glossyreflector spheres with roughness of 200, 1000 and 100000. You can still see quite a bit of graininess in the left sphere even though this image used 64 samples per pixel. Glossy surfaces require many more samples for high quality images. Also notice that there is no specular highlight from the infinite light source. Infinite lights don't represent real, physical entities. An area light would show a specular highlight however.

© 1999 - 2017 R. Steven Glanville



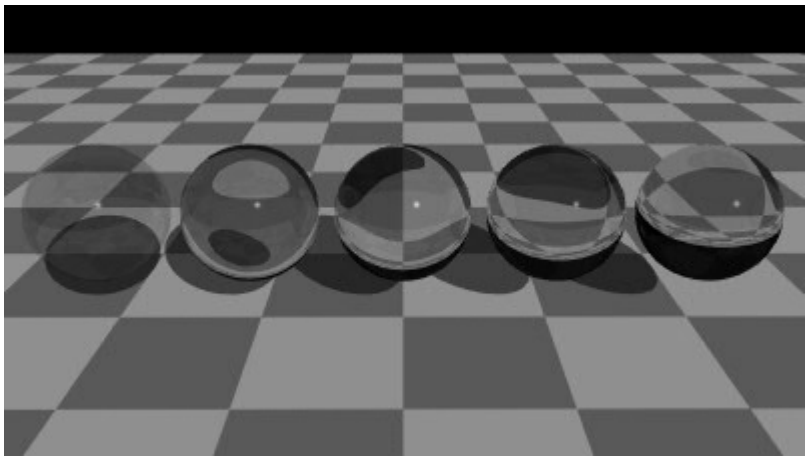
Class transparent

This transparent class is a simple model for diffraction of light through transparent materials. There is also a slight reflection off the surface. The amount of transparency is set with the normal transparency value for a material. It has one additional attribute:

- IOR - the index of refraction. This represents how much light is bent as it enters or leaves a transparent material. Useful values include 1.0 (air - no difference than outside of a material), 1.33 (glass), 1.49 (Plexiglas) and 2.42 (diamond).

Rendering transparent materials can take a very long time. Each primary ray (from the camera) can trigger tens or even hundreds of rays since each interaction generates up to two child rays, the reflected ray and the transparent ray.

The image below show 5 spheres with and IOR of 1.0, 1.1, 1.2, 1.3 and 1.5, and a transparency of 0.1:



Class dielectric

The dielectric class is a complex model that is closer to how light interacts with real transparent materials. It uses Fresnel equations to determine reflections and transmission parameters at each ray-surface interaction. As light travels through a dielectric material it is attenuated and gradually takes on the material's color. Thin sections appear almost colorless but thick sections can be quite dark. Glass is a dielectric which explains why a solid glass door looks clear from

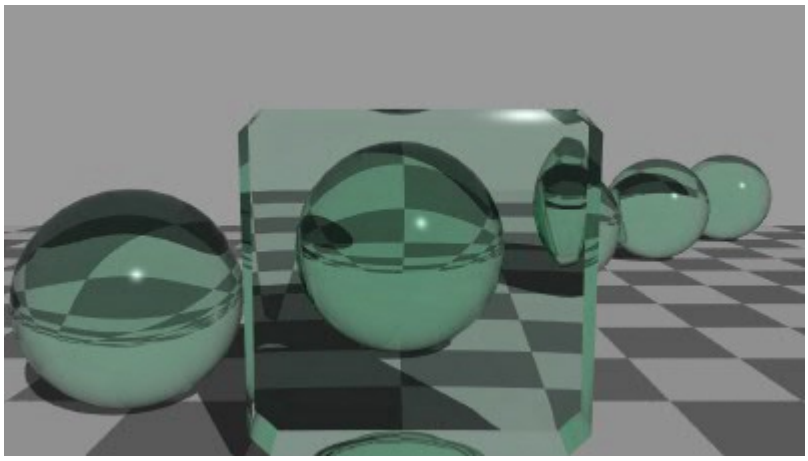
© 1999 - 2017 R. Steven Glanville

the front but looking into the edge can be dark green or gray. The dielectric class has two additional attributes:

- IOR - the index of refraction. As with the transparent class, this controls how much light bends as it enters or leaves the material.

- UnitDistance - the distance a ray must travel through the material to be filtered by the material's diffuse color. The ray's color is multiplied by the diffuse color raised to the power of (the distance traveled/UnitDistance). So if the diffuse color is (0.5, 0.5, 0.5) and UnitDistance is 10, then a ray that travels 20 units will be scaled by (0.5, 0.5, 0.5)^2 or (0.25, 0.25, 0.25). The default value of UnitDistance is 10.0.

The image below shows several spheres and a rectangular prism that are dielectrics with an IOR of 1.3 approximating glass:



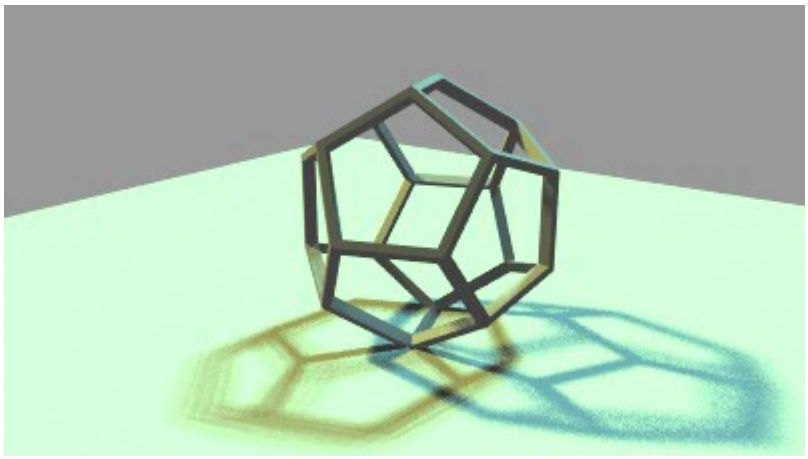
Light Attributes

You can modify the look of area lights by changing the way that they are sampled with the following attribute on the light in a scene:

- sampler - is a string attribute with three recognized values: multijittered (the default), jittered, and regular. The jittered modes produce a more random or grainy effect when under sampled while the regular sampler looks more like multiple shadows.

The image below has two area lights, the blue one (the yellow shadow) uses a regular sampler while the white light (which makes a blue shadow) uses the default multijittered sampler.

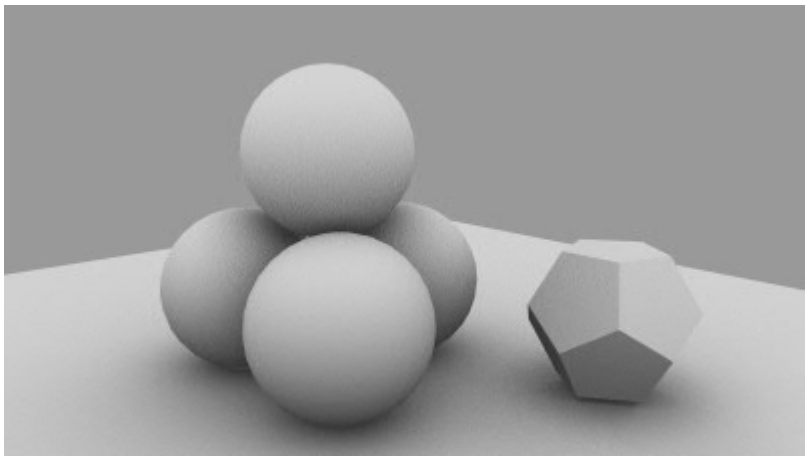
© 1999 - 2017 R. Steven Glanville



Ambient Occlusion

Ambient occlusion approximates light reflected from surfaces other than light objects. It adds a realistic aspect to soft shadows in corners and where objects meet.

You enable ambient occlusion in Anim8or by setting the Scene int attribute AmbientOccluder to 1. Together with the Global Lighting Parameters of **Global Intensity** and **Ambient Intensity** (accessible via the Settings→Lighting command) you can render realistic corner shadows such as shown below:



There are no lights in this scene. Only the color of the background was used as a source of light. The settings were: Global Intensity of 0.0 (equivalent to no lights used), Ambient Intensity of 3.14 (required for correct lighting using background only), the int Scene attribute AmbientOccluder with a value of 1, and 256 AA samples/pixel. Anti-aliasing must be enabled for Ambient Occlusion.

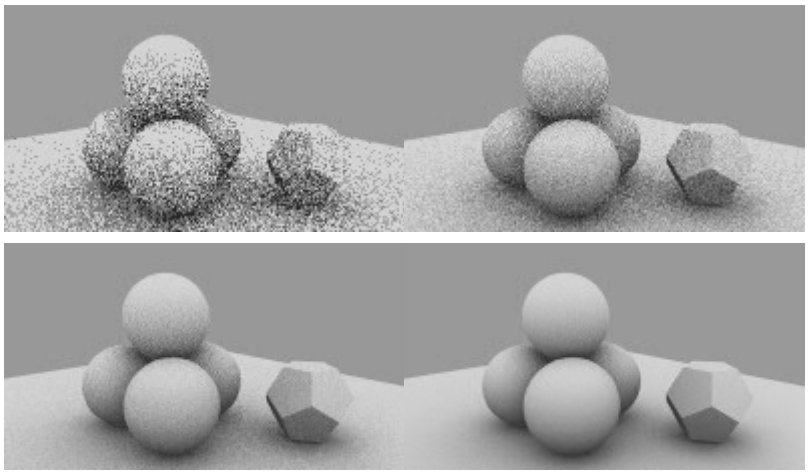
Ray Tracing and Anti-Aliasing

Ray tracing uses statistical sampling techniques to render global lighting, reflections, transparency, diffraction effects. High quality images require multiple sample rays per pixel. This, of course, means longer rendering times.

There is no single minimum number for a "good" image. You'll have to experiment to find what works for your scenes. Ambient occlusion using only the background color, as in the image

 © 1999 - 2017 R. Steven Glanville

above, can be particularly compute-intensive because light needs to be sampled from all directions. Here are three images of the same scene rendered with 4, 16, 64 and 256 samples:



© 1999 - 2017 R. Steven Glanville

Chapter 12 - Keyboard Shortcuts

Keyboard Shortcuts

There are numerous **keyboard shortcuts** that you can use to facilitate your modeling. They do not add new functionality. For every shortcut there is an equivalent toolbar button or menu command that you may use instead.

Commands that are available in more than one editing mode use the same shortcut. The control-style and most lower case shortcuts fall into this category. Some shortcuts perform different command in each editor. These are generally the uppercase ones.

You can also use shortcuts for most menu commands. The name of each shortcut is shown in the menu to the right of the command name.

Menu Shortcuts

The following are standard Windows shortcuts:

^N	New project
^O	Open project
^S	Save project
Sh+^S	Save all

Viewpoint Shortcuts

You can use the numeric keypad to switch the working view to any of the standard viewpoints.

7 Back View	**8** Top View	**9** Ortho View
4 Left View	**5** Front View	**6** Right View
1 Camera View	**2** Bottom View	**3** Perspective View
0 Previous View		**.** Toggle 1/4 views

Top Toolbar Shortcuts

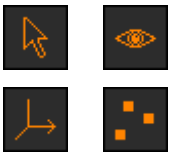
These shortcuts are for the top toolbar:

^Z	**^Y**	**^X**	**^W**	**^F**	**^U**	**^A**
Undo	Redo	Cut	Wire Frame	Flat Shaded	Smooth Shaded	Material Toolbar

^R	**^G**	**^T**	**^L**	**^P**	**^D**
Arc Rotate	Grid Snap	Fast Select	List Items	Graph Editor	CAS Annotations

© 1999 - 2017 R. Steven Glanville

Object Editor Shortcuts



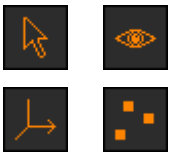
A	Edit mode	**V**	Viewpoint mode
O	Axis mode	**P**	Point edit mode

w	World coordinates	**o**	Object coordinates	**j**	Screen coordinates
x	X Axis	**y**	Y Axis	**z**	Z Axis

a	Select	**d**	Drag select
m	Move	**r**	Rotate
n	Non-uniform scale	**s**	Scale
l	Line	**p**	Path
U	UV editor		

S	Sphere	**C**	Cube
Y	Cylinder	**U**	Primitive mesh
G	N-gon	**T**	Text
W	Warp modifier		

Point Editor Shortcuts



A	Edit mode	**V**	Viewpoint mode
O	Axis mode	**P**	Point edit mode

w	World coordinates	**o**	Object coordinates	**j**	Screen coordinates
x	X Axis	**y**	Y Axis	**z**	Z Axis
v	Point	**e**	Edge	**g**	Face

© 1999 - 2017 R. Steven Glanville

[	Front	]	Back

a	Select	d	Drag select
m	Move	r	Rotate
n	Non-uniform scale	s	Scale
	Add edge		UV editor
X	Extrude faces	Y	Extrude faces connected
P	Peak faces	b	Bevel
I	Inset	u	Shell
B	Bump	C	Cut Faces
k	Topographical knife	S	Slide
E	Extrude edges		

Figure Editor Shortcuts

A	Edit mode	V	Viewpoint mode

w	World coordinates	o	Object coordinates	j	Screen coordinates
x	X Axis	y	Y Axis	z	Z Axis

a	Select	d	Drag select
m	Move	r	Rotate
s	Scale		

E	Edit bones	R	Rotate bone

© 1999 - 2017 R. Steven Glanville

L Bone length **N** New bone

S Skin object

X Show axis **B** Show bones

O Show objects **I** Show IK controls

Sequence Editor Shortcuts

A Edit mode **V** Viewpoint mode

w World coordinates **o** Object coordinates **j** Screen coordinates

x X Axis **y** Y Axis **z** Z Axis

a Select **R** Rotate bone

i Inverse kinematics

Ret Play **Space** Pause/Resume **Escape** Stop

X Show axis **B** Show bones

O Show objects **I** Show IK controls

G Show ghosts

K Animate

Scene Editor Shortcuts

A Edit mode **V** Viewpoint mode

w World coordinates **o** Object coordinates **j** Screen coordinates

© 1999 - 2017 R. Steven Glanville

x X Axis **y** Y Axis **z** Z Axis

a Select **d** Drag select

m Move **r** Rotate

s Scale **p** Edit path

R Rotate bone **i** Inverse kinematics

Ret Play **Space** Pause/Resume **Escape** Stop

X Show axis **B** Show bones

O Show objects **I** Show IK controls

O Show camera **I** Show path

K Animate

© 1999 - 2017 R. Steven Glanville