AXEL™

Tutorials

PN: TG-AXEL-v100

MINDAVENUE

Title: AXEL Tutorials

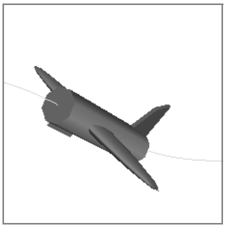
Version: 1.0

First published in on-line format, April 2001

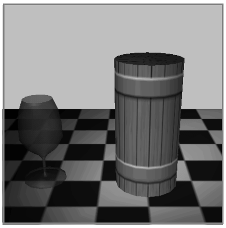
Contents

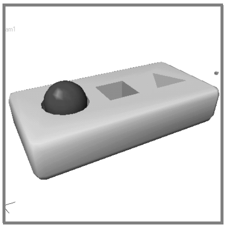
Introduction 9

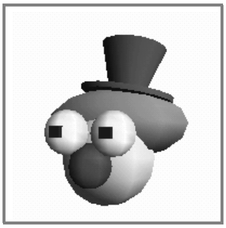
Fast Track 13

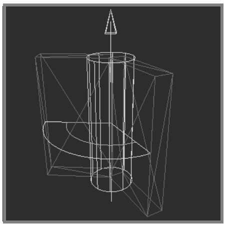


Airplane 21



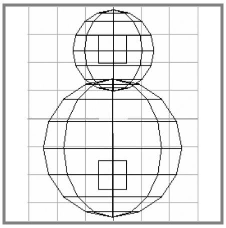




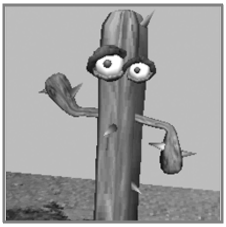


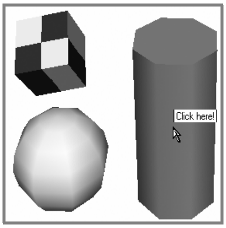
Pin, Ball, and Slider 65

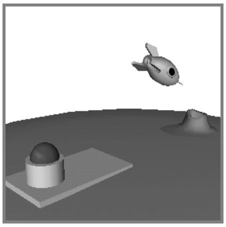
Furniture 77

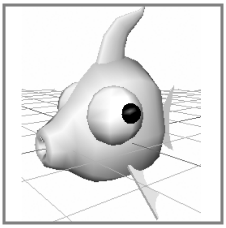


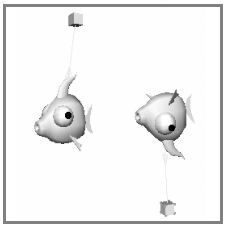
Snowman 85

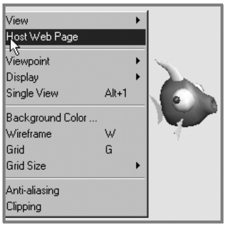



Click here!








View
Host Web Page
Viewpoint
Display
Single View Alt+1
Background Color ...
Wireframe W
Grid G
Grid Size
Anti-aliasing
Clipping

Learning AXEL

Introduction

Welcome to the AXEL tutorials. These lessons provide hands-on training in creating interactive 3D content for the web. The lessons vary in length and complexity. You can follow the lessons in order from lesson 1 to 13, or start with the Fast Track then select the next lessons according to your particular training needs.

In this introduction:

Prerequisites

You should start your AXEL training with the Jump Start lessons in the On-line help menu. When you have completed the Jump Start, you will be familiar with the interface and the most frequently used tools and hot keys allowing you to progress through the printed tutorials faster.

Which tutorials are for you?

These tutorials contain a Fast Track and 13 lessons.

If you are new to 3D modeling and animation, skip the Fast Track and start with Lesson 1, "Airplane" on page 21. In lesson 1, you will learn how to model, animate, add interaction, and publish an interactive 3D World. In the subsequent lessons you will discover more features, and learn how to create more intriguing Worlds.

If you have experience with 3D computer graphics, you should start with the Fast Track. The Fast Track and the hot key card may be all you need to transfer your 3D skills to AXEL; however, we recommend you do the following advanced lessons before experimenting on your own:

- Lesson 8, "Mr. Cactus" on page 93 covers IK and textures.
- Lesson 10, "Rocket Launch" on page 115 shows you how to add interaction, so that web surfers can manipulate objects and trigger animations, sounds and other reactions.
- Lesson 11, "Fish — Modeling" on page 129 includes revolutions, extrusions, and shape editing.
- Lesson 12, "Fish — Animating" on page 137 includes interactive recording, path animation, and animation curve editing.
- Lesson 13, "Fish — Publishing" on page 153 shows you how to publish your World in an existing web page, export parameters and integrate the parameter controls with 2D elements using JavaScript.

Documentation conventions

The AXEL documentation uses the following conventions:

Menu commands

The forward arrow indicates menu selections. For example, *Choose File* > *Save* means click the File menu, then choose the Save command from the menu.

Hot Keys

Hot keys appear in bold and small caps type. For example, press **T**; press **SHIFT+Q**; hold **L**.

Press **T** means press and release the T key.

Press **SHIFT+Q** means hold down the Shift key, press and release the Q key, then release the Shift key.

Hold **L** means press and hold down the L key. When you release the key, AXEL returns to its previous state.

For a complete list of available hot keys, refer to the AXEL hot key card.

Additional learning material

The following training material is also provided to help you master AXEL quickly.

Jump Start	These lessons, in the AXEL Help menu, introduce basic 3D concepts and give an overview of the AXEL interface and workflow. This is a good starting place to finding your way around AXEL.
AXEL User Guide	The User Guide describes all the features and shows you how to use them through step by step procedures and examples.
On-line Help	AXEL On-line Help contains the same information as the user guide. In addition, it contains hyperlinks and search features to help you find answers fast.
Resources on www.MindAvenue.com/ resources/	Here you'll find FAQs, the AXEL forum including tips and tricks, and the Request Form for additional support. You will also find tutorials that were added after the printing of this manual.
AXEL Hot Key card	The hot key card lists all the hot keys available in AXEL and includes a quick reference to the AXEL workspace.

Note: The user guide and tutorials are also available in PDF format on your software CD.

AXEL basics for experienced 3D artists

Fast Track

The Fast Track is designed for those who are already familiar with 3D animation software. It provides a quick reference section and a 10-minute tutorial.

If you are new to 3D, go straight
to Lesson 1, “Airplane” on page 21.

In this lesson:

Quick reference

This section contains the minimum of what you need to know about AXEL to get started fast. It covers frequently used tools, and concepts that may not be common to other 3D software.

Author and Browse modes

In *Author* mode, you create and modify your 3D content. To test the content, you switch to *Browse* mode. In Browse mode, the WebCam view turns into a preview of what the content will be on the Internet; you can preview animations and test interactions.

To switch between Author and Browse mode:

Press **CTRL+B** or click the Author ( ) or Browse ( ) button.

Play

Some effects and deformations will not appear in the viewports unless time is running. You can always stop the playback by clicking on the Play button (▶) or pressing the **P** hot key.

Tool modes

There are 3 tool modes: Object, Point, and Center. These modes work in conjunction with the selection and manipulation tools so if you are trying to select points and nothing is happening, make sure you are in point mode.

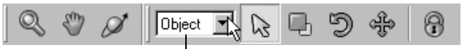
To select a tool mode:

Do one of the following:

- Press one of the following hot keys:

3	Object mode
4	Point mode
5	Center mode

- In the toolbar, select a mode from the list.



Mode list

Selection/manipulation tools

Tools work in conjunction with modes. The default tool is Select ( ), if no other tool is active, AXEL activates the Select tool. You can also select objects using the manipulation tools, Scale ( ), Rotate ( ), and Translate ( ); the corresponding hot keys are **E**, **R**, **T**. (These have been mapped ergonomically to be close to the mode hot keys). Pressing a hot key activates the tool; holding a hot key activates the tool and locks the current selection. **SHIFT**-click and drag with the left, middle or right mouse button to limit the manipulation to the X, Y, or Z axis respectively.

Example, to translate an object:

- Select the object, press **T**, then click and drag in the viewport.

- To avoid accidentally changing the selection, hold the **T** hot key while translating the object.

- To translate along the Y axis, **SHIFT**-drag with the middle mouse button.

Multi-selection

You can select objects in the World Explorer or in the viewports.

To select objects:

- Click an object to select it.

- **SHIFT**-click objects to add them to the selection.

- **CTRL**-click objects to remove them from the selection. The **CTRL** key toggles the selection.

Note: The order of selection is important when multi-selecting to apply a command. The first object is always the one that will be controlled, the second (or subsequent) is the controlling object.

Unlocking constraints

The unlock button ( ); temporarily unlocks constraints so that you can manipulate objects that are constrained to other objects. It is mapped to the **U** hot key.

To unlock constraints:

Use the unlock button or hot key:

- Hold **U** while manipulating the object in the viewports. When you release the hot key, the objects are constrained again.

- Click the Unlock button then manipulate the objects. Click the Unlock button again to lock the constraints.

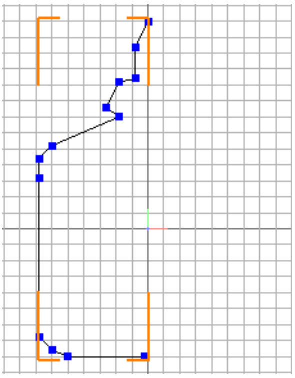
From modeling to publishing in 10 minutes

In this tutorial, you will create a 3D object with a texture, animation, interaction, and then publish the file for the internet.

Sine Us!
The Spray to inhale.

Model a sinus spray bottle

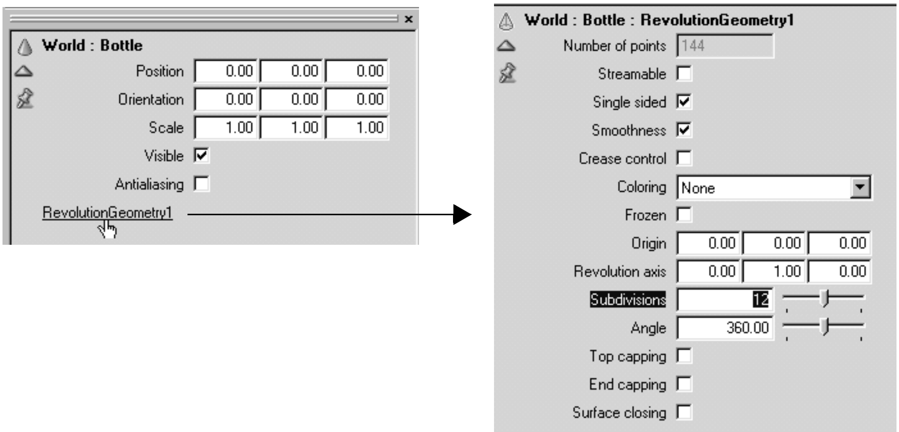
1. Launch AXEL.
2. Draw the profile of a bottle:
   a) Choose Modeling > Draw Curve (press and hold the **L** hot key), and click points in the Front viewport to draw the profile of a bottle.
   b) To turn on Snap to Grid, choose View > Snap to Grid.



3. Create a revolution from the curve:
   a) With the curve selected, choose Modeling > Create Surface > Revolution from Curve.
   b) Rename the revolution surface. Select it in the World Explorer then click it a second time, type *Bottle* and press **ENTER**.

4. Modify the profile curve.
   a) Select the curve, switch to Point mode (press the **4** hot key), and translate points on the curve.
      Notice how the revolution surface is updated accordingly.
   b) You can also add points to the curve by choosing Modeling > Modify Curve > Add Point (press the **INS** hot key), then click on the curve to add a point.
   c) When you are satisfied with the shape of the curve, select it and turn off its visibility in the Parameter Editor.

5. Increase the subdivisions on the revolution.
   a) Switch back to Object mode (press **3**), and select the bottle.
   b) In the Parameter Editor, click RevolutionGeometry, then increase the subdivisions to 12.



World : Bottle

Position 0.00 0.00 0.00

Orientation 0.00 0.00 0.00

Scale 1.00 1.00 1.00

Visible ☑

Antialiasing □

RevolutionGeometry1

World : Bottle : RevolutionGeometry1

Number of points 144

Streamable □

Single sided ☑

Smoothness ☑

Crease control □

Coloring None

Frozen □

Origin 0.00 0.00 0.00

Revolution axis 0.00 1.00 0.00

Subdivisions 12

Angle 360.00

Top capping □

End capping □

Surface closing □

Add a texture

1. Select the bottle and choose Visualization > Add Texture. Navigate to the Documentation / Tutorials / Textures folder and select *Sineus.bmp*.

2. Choose Visualization > Texture Projection > Cylindrical.

   The texture projector appears as a wireframe cylinder in the viewports.

3. Scale the texture projector so that the texture appears correctly on the bottle, then delete the texture projector when you are happy with the placement of the texture.

4. In the World Explorer, expand Bottle / Material and select Texture Settings, then turn off Horizontal and Vertical tiling in the Parameter Editor.

Add the spray

1. Place a sphere at the tip of the bottle.
   a) Choose Modeling > Preset Surfaces > Sphere.
   b) Translate the sphere so that its center is at the tip of the spray bottle.

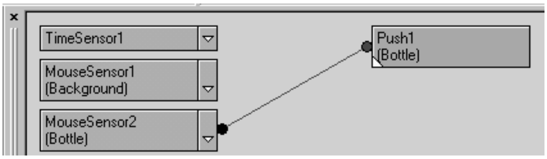
2. With the sphere still selected, choose Animation > Add Particles.

Note: The Play button must be on for the particles to be visible.

3. Select the sphere and turn off its visibility in the Parameter Editor.

Add interaction:

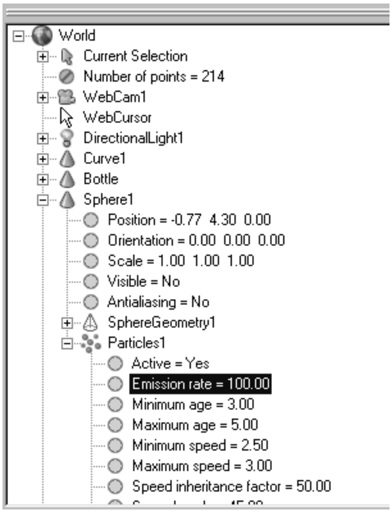
1. Add a Push handle. Select the bottle and choose Interaction > Handles > Push.
   In the Interaction Editor, notice the mouse sensor on the bottle linked to the Push handle.



When a web surfer clicks the bottle, it will be pushed in. Switch to Browse mode (press **CTRL+B**) to test the interaction. Switch back to Author mode (press **CTRL+B**).

2. Add a reaction to control the particles:
   a) In the World Explorer, expand the Sphere and select Particles.
   b) In the Parameter Editor, set the Emission rate to 0.
      The particles stop flowing.

c) Select the Emission rate parameter, and choose Interaction > Reactions > Toggle Parameter.

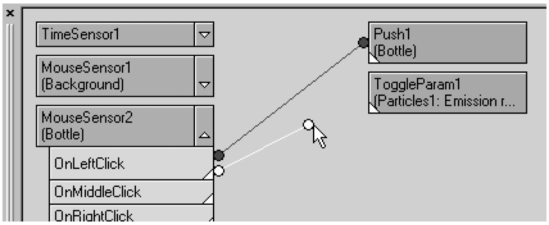


World
Current Selection
Number of points = 214
WebCam1
WebCursor
DirectionalLight1
Curve1
Bottle
Sphere1
Position = -0.77 4.30 0.00
Orientation = 0.00 0.00 0.00
Scale = 1.00 1.00 1.00
Visible = No
Antialiasing = No
SphereGeometry1
Particles1
Active = Yes
Emission rate = 100.00
Minimum age = 3.00
Maximum age = 5.00
Minimum speed = 2.50
Maximum speed = 3.00
Speed inheritance factor = 50.00

The Interaction Editor now shows two reactions.

d) Select the ToggleParam1 reaction, and adjusts its parameters in the Parameter Editor. Set the Start value to 0 and the Stop value to 100.

e) In the Interaction Editor, connect the ToggleParam reaction to the mouse sensor on the bottle. In the Interaction Editor, expand MouseSensor (Bottle), and click and drag from the OnLeftClick to the ToggleParam reaction.



f) Select the trigger (the line connecting the sensor and the reaction), then, in the Parameter Editor, set the Trigger on field to Hold.
When the web surfer holds the mouse over the bottle, the Emission rate parameter will toggle from 0 to 100.

3. Add a sound.

a) Select the Bottle and choose Interaction > Reactions > Play Sound, navigate to Documentation / Tutorials / Sounds, and select *steam.wav.*

4. Add camera interaction. Choose Interaction > Handles > Orbit WebCam.

Test the content

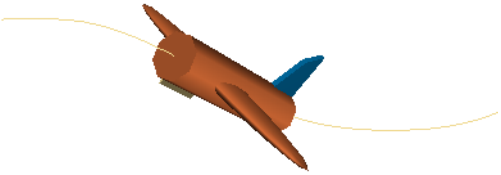
1. Switch to Browse mode (press **CTRL+B**), and click on the bottle.
2. Switch back to Author mode (**CTRL + B**) to continue working on the content.

Publish your work

1. Save the Project file (.axel). Choose File > Save, and select the name and location of your choice. You can reopen the project file at any time to continue working on the project.
2. Publish a stream file in a host web page. The stream file contains all the content compressed and ready to playback in a web page.
   a) Choose Web Integration > Publishing Settings.
   b) Enter a name for the stream file (.axs).
   c) Enter a name for the host web page file.
   d) Click Publish.
   e) Choose Web Integration > Preview in IE and test your World.

1 Modeling, Animating, Publishing
Airplane

In this lesson, you build a simple airplane using default surfaces and an extrusion, which you then group together. You animate the group along a path, then animate one of the group's components. Finally, you publish the plane so it flies over the background of a web page.



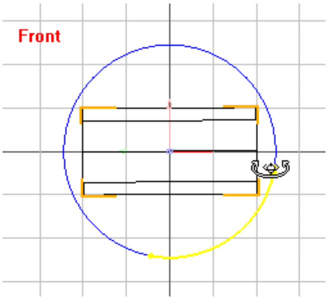
Need help? — Refer to the *Plane_complete.axel* file in the AXEL / Documentation / Tutorials folder.

In this lesson:

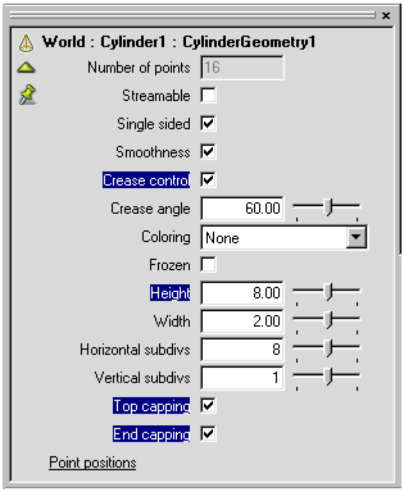
Creating the plane from preset surfaces and extrusions

Build a plane using default surfaces and an extrusion that you translate, scale, and rotate into shape. Group the objects to make it easier to add animation and constraints.

1. Create the fuselage:
   a) Choose Modeling > Preset Surfaces > Cylinder.
   b) In the Front viewport, rotate the cylinder 90 degrees so it is horizontal.



   c) With the cylinder still selected, click CylinderGeometry1 in the Parameter Editor, and set the following parameters:



World : Cylinder1 : CylinderGeometry1

Number of points 16

Streamable □

Single sided ☑

Smoothness ☑

Crease control ☑

Crease angle 60.00

Coloring None

Frozen □

Height 8.00

Width 2.00

Horizontal subdivs 8

Vertical subdivs 1

Top capping ☑

End capping ☑

Point positions

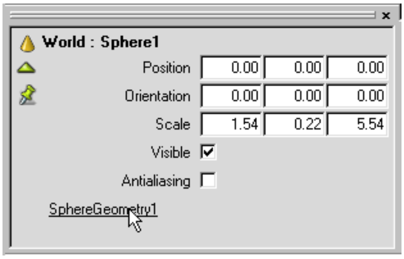
2. Create the wings:
   a) Choose Modeling > Preset Surfaces > Sphere.

b) Use the Top and Right viewports to scale the sphere into wings.

Use:	To:
Left mouse button	Scale proportionally.
Middle mouse button	Scale in the plane of the viewport.
SHIFT+left, middle, or right mouse button	Scale in one axis at a time.

Note: For more information on the Scale tool, choose Help > On-Line Help, and look up Modeling / Selections and Transforms / Scaling Objects.

c) With the sphere still selected, click SphereGeometry1 in the Parameter Editor.



World : Sphere1

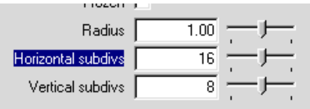
Position	0.00	0.00	0.00
Orientation	0.00	0.00	0.00
Scale	1.54	0.22	5.54

Visible ☑

Antialiasing □

SphereGeometry1

d) Set Horizontal subdivisions to16.



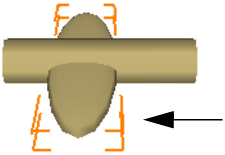
Radius 1.00

Horizontal subdivs 16

Vertical subdivs 8

The wing's surface is now smoother.

e) Click the Translate button (✥) and use the **ARROW** keys to nudge the wings closer to the front of the plane.



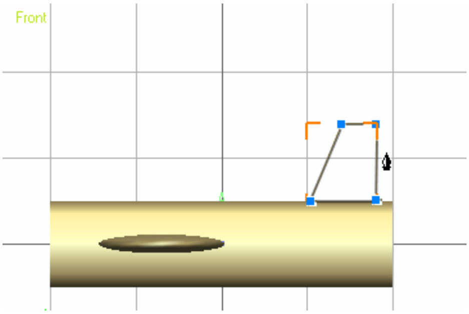
The **ARROW** keys translate an object by a set number of pixels. If you zoom in up close, you are nudging by very small increments; if you zoom out, you are nudging by larger increments.

3. Create the tail:

   a) Choose Modeling > Draw Curve.

   b) Choose View > Snap to Grid (**SHIFT+H**).
   The Snap to Grid function will force the cursor to snap on an invisible grid, 1/5 the size of your actual grid settings. This is practical for drawing straight lines. It also allows you to move objects in precise increments.

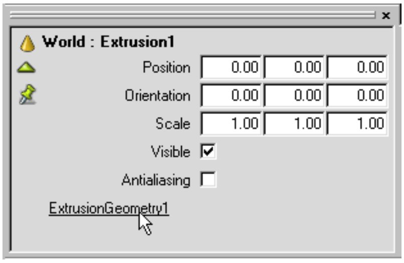
   c) In the Front viewport, draw the first four points of the tail and press **END** to close the curve.



   d) When satisfied with the result, toggle off Snap to Grid.

   e) Choose Modeling > Create Surface > Extrusion from Curve.
   A surface is extruded from the curve.

   f) With the extrusion still selected, click ExtrusionGeometry1 in the Parameter Editor.



World : Extrusion1

Position 0.00 0.00 0.00

Orientation 0.00 0.00 0.00

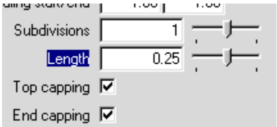
Scale 1.00 1.00 1.00

Visible ☑

Antialiasing ☐

ExtrusionGeometry1

   g) Set the Length to 0.25.
   Now the tail is not as thick.



Subdivisions 1

Length 0.25

Top capping ☑

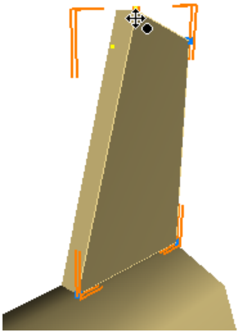
End capping ☑

4. Adjust the shape of the tail:

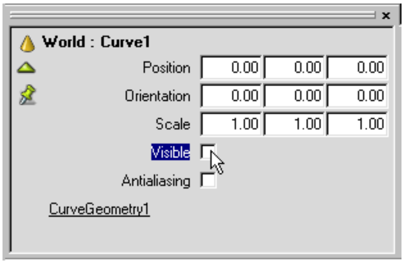
   a) In the World Explorer, select the curve.

   b) Switch to Point mode (press **4**).

   c) Select the tail's points and translate them in the Front viewport.
   As you change points on the original curve, the extrusion follows.



   d) Once you like the shape, switch to Object mode (press **3**) and delete the curve or hide it by turning off Visible in the Parameter Editor.



World : Curve1

Position	0.00	0.00	0.00
Orientation	0.00	0.00	0.00
Scale	1.00	1.00	1.00
Visible	□		
Antialiasing	□		

CurveGeometry1

5. Group the different parts of the plane:

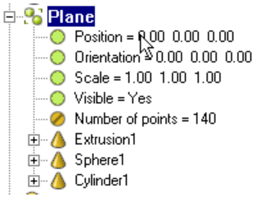
   a) Multi-select the cylinder, sphere, and extrusion. **SHIFT**-click each object in the World Explorer.

   b) Choose Relations > Group.
   A group is created. You can translate, scale, rotate, animate and add relations and interactions to a group just like you can with any other object.

6. Rename the group:

   a) In the World Explorer, select the group and click it again.
   The name is highlighted.

b) Type *Plane* and press **ENTER**.

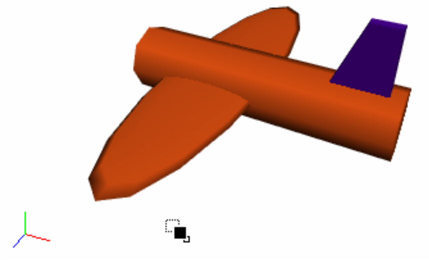

Plane
Position = 0.00 0.00 0.00
Orientation = 0.00 0.00 0.00
Scale = 1.00 1.00 1.00
Visible = Yes
Number of points = 140
Extrusion1
Sphere1
Cylinder1

7. Add color to the plane:

   a) With the Plane group still selected, choose Visualization > Add Material.

   b) Pick a color and click OK.
   All objects in the group now share the selected color (Material1).

   c) Expand the Plane group, select the extrusion, and choose Visualization > Unlink Material**.**
   The tail returns to the default color. Now give the tail its own color.



8. Choose File > Save As to save your project.

Adding a landing gear and moving its center

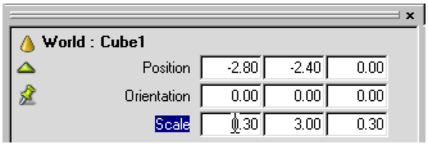
Create the landing gear, add it to the group, change its pivot point, then animate it.

1. Create the landing gear:

   a) Choose Modeling > Preset Surfaces > Cube.

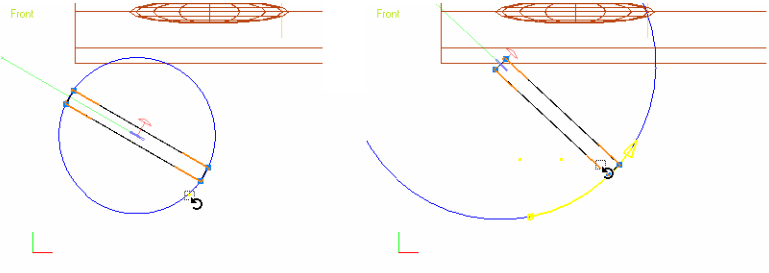
   b) Move the cube towards the front of the plane, under the cylinder.

   c) Scale the cube so it is rectangular. Type the following values in the Parameter Editor, pressing **TAB** to go from cell to cell.


World : Cube1

Position	-2.80	-2.40	0.00
Orientation	0.00	0.00	0.00
Scale	0.30	3.00	0.30

2. Move the cube's center so it can fold into the plane after takeoff. Leaving the center in its default position, the middle of the cube, will make it pivot around that point (left). By moving the center to the top of the cube, the cube will rotate around its top (right).



   a) With the cube selected, switch to Center mode (press **5**) and translate the center to the top of the cube.

   b) Switch to Object mode (press **3**), and rotate the cube in the Front viewport.
   The cube rotates around its center and folds into the plane.

3. Attach the landing gear to the plane:

   a) In the World Explorer, select the Plane group.

   b) Press **SHIFT** and select the cube. It is important that you first select the group, then multi-select the object you want to add to the group.

   c) Choose Relations > Add to Group.
   The cube is added to the Plane group. (You can add or remove any number of objects to and from a group.)

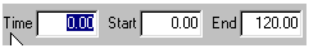
Animating the landing gear's orientation

Create two keys to make the landing gear move from one position to another over a period of 5 seconds.

1. At time 0, set a key with the landing gear down:

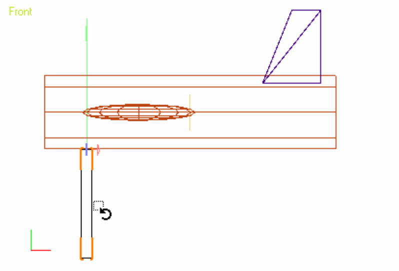
   a) Turn off the Play button ▶ or press **P** to stop playback.

   b) Set the current time to 0.

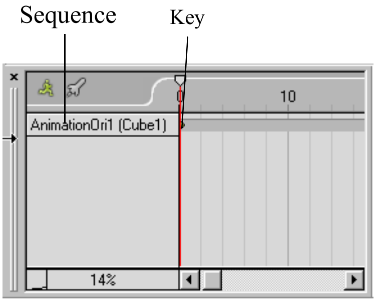


Time 0.00 Start 0.00 End 120.00

c) Rotate the cube so that the landing gear is down, ready for landing.

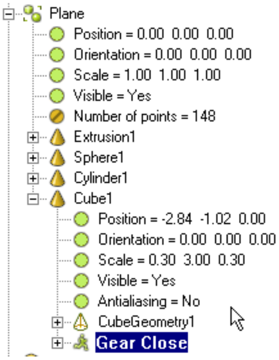


d) With the cube selected and in Rotate mode, turn on the Record button ● .
A key and an animation sequence appear in the Sequencer.


Sequence

Key

e) In the World Explorer, expand the landing gear cube and rename the animation *gear close*.



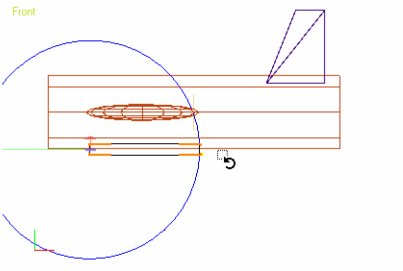
- Plane
  - Position = 0.00 0.00 0.00
  - Orientation = 0.00 0.00 0.00
  - Scale = 1.00 1.00 1.00
  - Visible = Yes
  - Number of points = 148
  - Extrusion1
  - Sphere1
  - Cylinder1
  - Cube1
    - Position = -2.84 -1.02 0.00
    - Orientation = 0.00 0.00 0.00
    - Scale = 0.30 3.00 0.30
    - Visible = Yes
    - Antialiasing = No
    - CubeGeometry1
    - Gear Close

2. Set another key with the landing gear up:

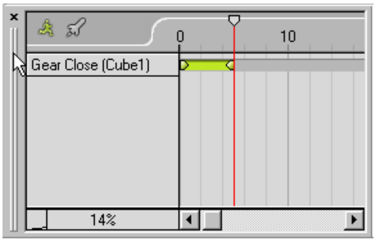
a) Go to time 5.00.

b) Make sure the Record button is still on.

c) Rotate the cube in the Front viewport.



A second key appears in the Sequencer.

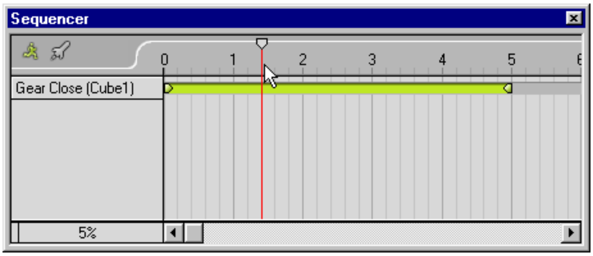


3. Refine the animation:

   a) Turn off Record, go to time 0.

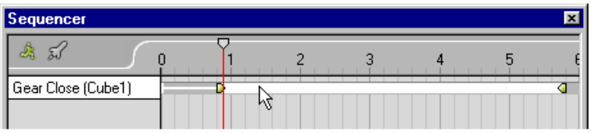
   b) To preview the animation, turn on the Play button or press **P**.

   c) Press **P** again to stop playback and drag the playhead to scrub through the animation.

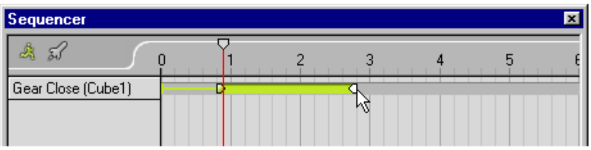


d) You can change the animation by dragging the sequence to offset it in time or by dragging a key in time.

Dragging the sequence



Dragging a key



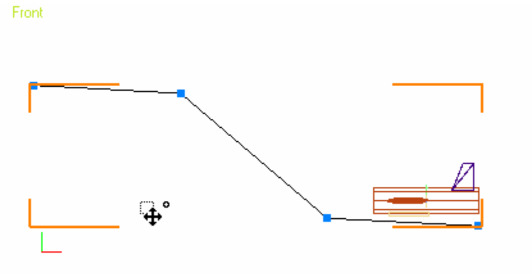
Note: For more information on animation, choose Help > On-Line Help, and look up Animation / Fine-tuning Animation.

Animating the plane along a path

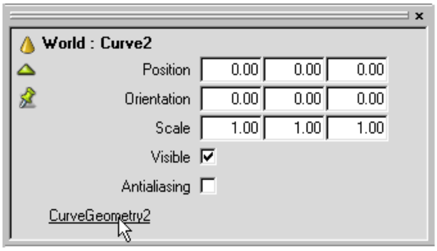
Draw a path, then animate the plane along that path.

1. Draw the path:
   a) Click the Front viewport to make it active, then press **ALT+1** to switch to Single View. Zoom out to make room to draw the curve.
   b) Choose Modeling > Draw Curve.
   c) Click several points to draw a curve. Press **ESC** twice to quit Draw Curve mode.

Note: When you press **ESC** once, you remain in Draw mode and can draw a new curve.



d) With the curve selected, click CurveGeometry1 in the Parameter Editor.



World : Curve2

Position 0.00 0.00 0.00

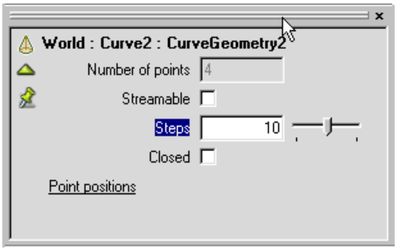
Orientation 0.00 0.00 0.00

Scale 1.00 1.00 1.00

Visible ☑

Antialiasing □

CurveGeometry2



World : Curve2 : CurveGeometry2

Number of points 4

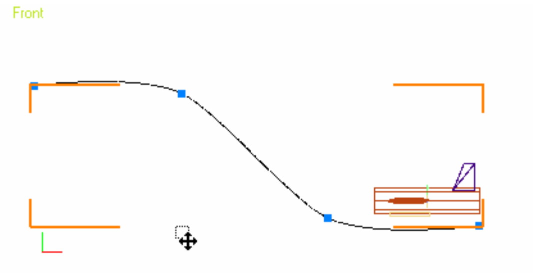
Streamable □

Steps 10

Closed □

Point positions

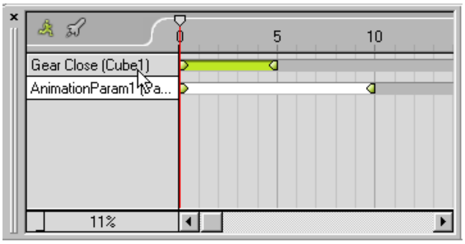
e) Set Steps to 10.
Increasing smoothness softens the curve without adding points.



2. Animate the plane along the path:

a) In the World Explorer, multi-select the Plane group and then the curve, in that order. It is important to first select the plane, then the curve.

b) Choose Animation > Path Animation.
The plane is repositioned on the path and an animation sequence is added to the Sequencer.



Gear Close (Cube1)

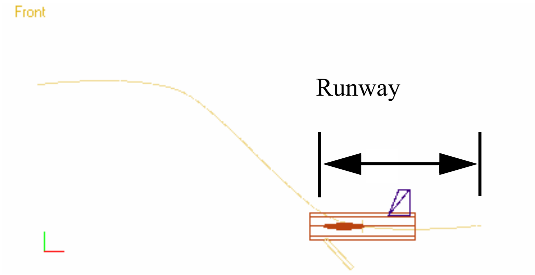
AnimationParam1 (Pa...

c) Drag the playhead to preview the animation.
The plane travels from the beginning to the end of the curve in 10 seconds.

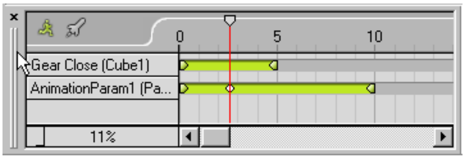
d) The plane travels on the curve between the first point you drew to the last. Depending on where you started drawing the curve, the plane may be flying backward. To make the plane fly in the opposite direction, select the curve and choose Modeling > Modify Curve > Invert.

3. View the animation:
   a) Switch to Multi View (press **ALT+1**). In the WebCam viewport, press **A** to frame all the objects.
   b) Click Browse to view the animation.
      The landing gear folds into the plane over a period of 5 seconds. The plane takes 10 seconds to complete its path.
   c) Click Author to continue working on the animation.

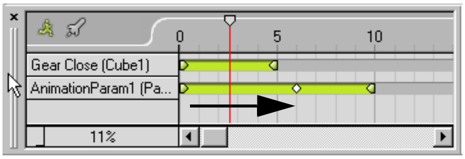
4. Adjust the timing:
   a) Drag the playhead to position the plane just before takeoff, at the end of the runway.



   b) Select the animation sequence, and choose Animation > Add Key.
      A new key appears in the Sequencer.

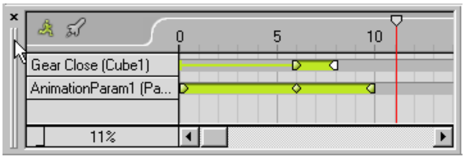


   c) Drag the key to time 6.



Now it takes the plane 6 seconds to get to the end of the runway and 4 seconds to fly over the rest of the path.

d) Make the landing gear extend until the end of the runway, then retract 2 seconds after takeoff. Drag the end of Gear Close to time 2, then drag the entire sequence so it starts at time 6.

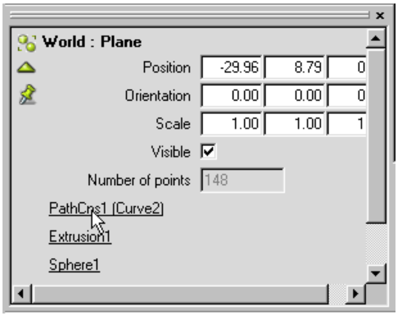


e) Drag the playhead to preview the result.

5. Make the plane follow the slope of the path instead of staying parallel to the ground:

a) Select the Plane group.

b) To view the constraint parameters, click PathCns1(Curve2) in the Parameter Editor.



World : Plane

Position -29.96 8.79 0

Orientation 0.00 0.00 0

Scale 1.00 1.00 1

Visible ☑

Number of points 148

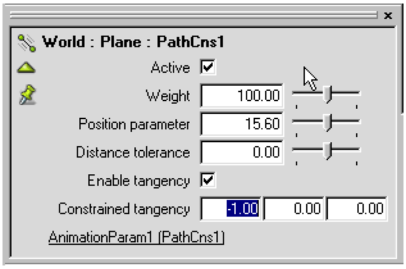
PathCns1 (Curve2)

Extrusion1

Sphere1

c) Turn on Enable tangency.

d) If the plane is upside down, change Constrained tangency to -1 0 0.



World : Plane : PathCns1

Active ☑

Weight 100.00

Position parameter 15.60

Distance tolerance 0.00

Enable tangency ☑

Constrained tangency -1.00 0.00 0.00

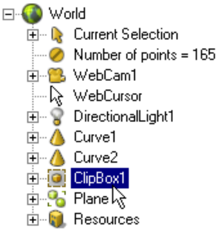
AnimationParam1 (PathCns1)

Constrained tangency determines which axis is tangential to the curve. Now when you play the animation, the plane follows the path in both position and orientation.

Adding a clip box

When you publish the plane, you will use a transparent background so that the underlying web page is visible. This is called *windowless publishing*. It allows you to layer 3D content with the 2D elements on the web page, but it can slow down the refresh rate of the web page. For windowless publishing, it is important to optimize the performance of the AXEL World as much as possible. Adding a clip box is one way to improve performance. Only the content inside the clip box is published. By limiting the amount of content that the computer has to draw, you improve the performance of the web page.

1. Create a clip box:
   a) Choose Visualization > Add ClipBox.
      A clip box appears in the World Explorer.



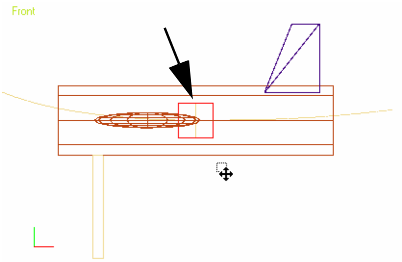
2. Choose View > Clipping to turn off clipping.

3. Choose View > Display > Clipbox to make the red clip box visible in all viewports.

4. Constrain the clip box to follow the plane so that the plane is always visible:

   a) In the World Explorer, select the clip box and then press **CTRL** and select the Plane group. When adding a constraint, it is important to first select the follower (the clip box) then multi-select the leader (the plane).

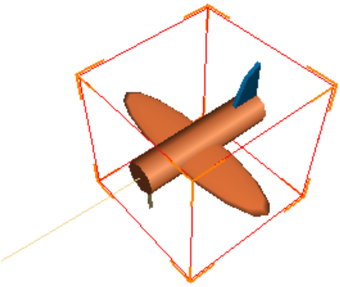
   b) Choose Relations > Pose Constraint.
      The clip box is constrained to follow the plane. It moves to the center of the plane, but is too small.



5. Adjust the size of the clip box:
   a) The pose constraint ties the clip box's position, scale, and orientation to the plane. To scale the clip box, you must disable the constraint. Turn on Unlock 🔓 to disable constraints.
   b) Scale the clip box.

   Note: You can also **CTRL**-drag to disable the constraint while you scale the plane.

   c) Check the Top, Front and Right viewports to ensure that the plane is in the clip box.



   d) Turn on the Lock button to lock the constraint.
   e) Hide the path. Select the curve in the World Explorer, and turn off Visible.
   f) In the WebCam viewport, zoom out to see the entire animation.
   g) Turn on Play to preview your work.
      The plane moves along the path and remains inside the clip box.

Publishing the plane

Publish the plane on a host web page. This step illustrates how you can layer 3D and 2D content on a web page.

1. Save the plane on a sample host web page:
   a) Choose File > Save As and save your World in the AXEL / Documentation / Tutorials folder.
   b) Choose Web Integration > Publishing Settings.
      Notice the publishing directory is set to the same folder as the saved project file.
   c) In the Stream file field, type *plane.axs*.
   d) In the Host web page field, type *plane.htm*.

e) Click Publish.
Two files are saved:

plane.axs	An AXEL stream file.
Plane.htm	A sample web page in which the AXEL stream file is embedded.

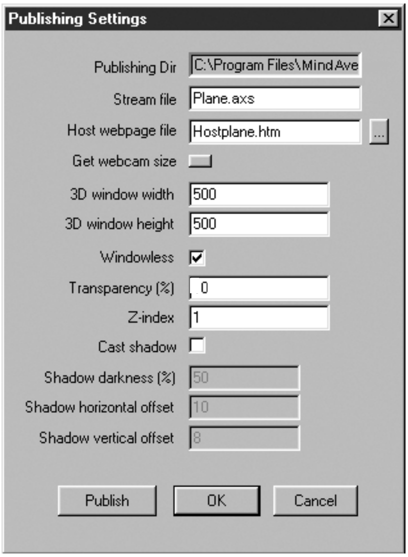
f) To view the result, right-click the WebCam viewport and turn on Host Web Page. To preview the page in a web browser, choose Web Integration > Preview in IE or Preview in Netscape.

2. Another web page, *Hostplane.htm*, was created for this lesson. You can add your 3D content to the 2D layers on this page.

a) Choose Web Integration > Publishing Settings.

b) Click the button next to the Host web page file field, navigate to Documentation / Tutorials, select *Hostplane.htm* and click Open.

c) In the Publishing Settings dialog, type the following settings:



Publishing Settings

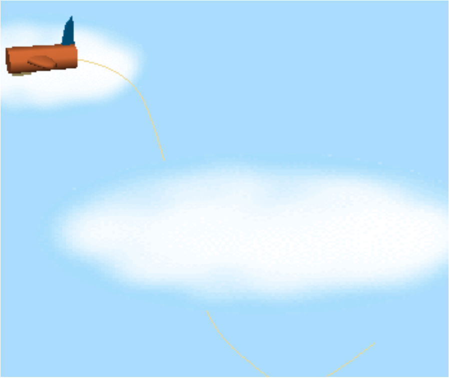
Publishing Dir	C:\Program Files\MindAve
Stream file	Plane.axs
Host webpage file	Hostplane.htm
Get webcam size	
3D window width	500
3D window height	500
Windowless	☑
Transparency (%)	0
Z-index	1
Cast shadow	□
Shadow darkness (%)	50
Shadow horizontal offset	10
Shadow vertical offset	8

d) Click Publish.
The AXEL stream *plane.axs* is incorporated in the *Hostplane.htm* page.

3. Adjust the 3D content so that it fits into the page design:

a) Drag the playhead to see the plane fly through the clouds on the Host web page viewport.

b) Adjust the path so that the plane flies from the bottom of the big cloud to the middle of the small cloud.



4. When you are satisfied with the result, choose Web Integration > Publish.

   The *Hostplane.htm* file is updated.

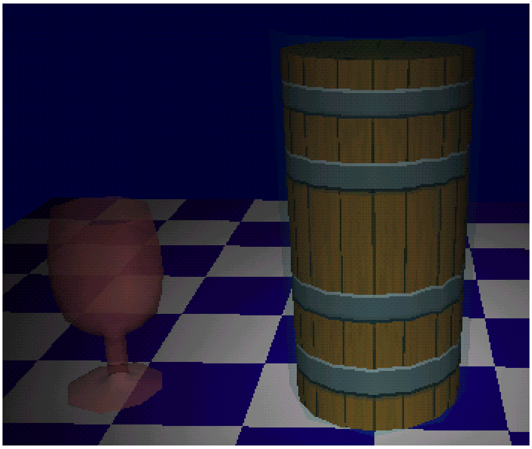
5. Open *Hostplane.htm* in a web browser and review your work.

   The plane flies behind the big cloud and in front of the small cloud. This is achieved by using the z-index parameter in the HTML file.

2 Visualization

Glass and Barrel

In this lesson, you create objects, add materials, then animate the World using a light and WebCam. As a finishing touch, you add an object interaction that produces a sound.



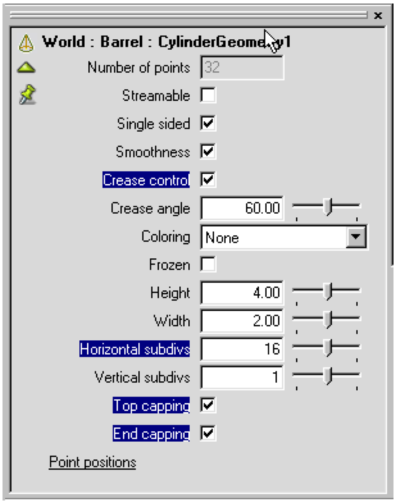
Need help? — Refer to the *Visualization_completed.axel* file in the AXEL / Documentation / Tutorials folder.

In this lesson:

Creating the objects

Create three objects, each based on a different modeling tool: a cylinder, a horizontal plane, and a revolution.

1. Use a cylinder for the barrel:
   a) Choose Modeling > Preset Surfaces > Cylinder.
   b) In the World Explorer, click Cylinder twice and rename it *barrel*.
   c) Expand the barrel and click CylinderGeometry1.
   d) In the Parameter Editor, set the following parameters.



World : Barrel : CylinderGeometry1

Number of points 32

Streamable □

Single sided ☑

Smoothness ☑

Crease control ☑

Crease angle 60.00

Coloring None

Frozen □

Height 4.00

Width 2.00

Horizontal subdivs 16

Vertical subdivs 1

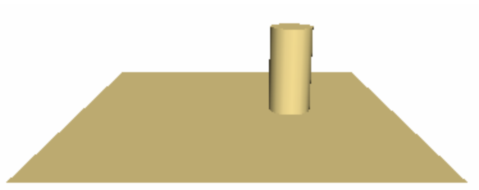
Top capping ☑

End capping ☑

Point positions

   e) Translate the cylinder to the right to make room for the other objects.

2. Use a horizontal plane for the tabletop:
   a) Choose Modeling > Preset Surfaces > Horizontal Plane.
   b) Scale the plane to 1.5 in X and Z.
   c) Translate the barrel on its Y axis until it sits on the plane.

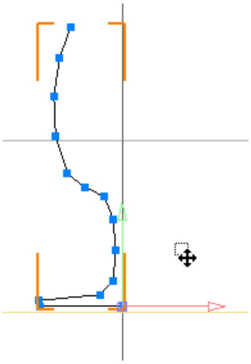


3. Draw the outline of a glass:

   a) Choose Modeling > Draw Curve, or hold the **L** key.

   b) In the Front viewport, draw the profile of a wine glass. Press **ESC** twice to end Draw Curve mode.

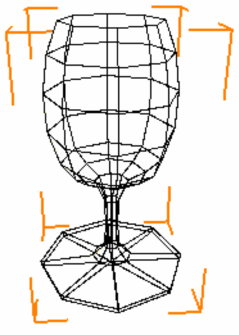
Tip: When you create the revolution, the profiles will spin around the Y axis, so you should draw the stem close to the Y axis but not cross it. Draw only its outline; do not give it thickness.



   c) Rename the curve *Glass_curve.*

4. Use a revolution to create a glass from the outline:

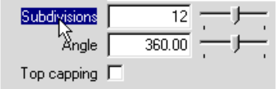
   a) With Glass_curve still selected, choose Modeling > Create Surface > Revolution from Curve.
   The outline spins around the World's Y axis to create a surface.



   b) Rename the Revolution Surface *Glass_surface.*

   c) To adjust the size of the glass, select Glass_curve and move it left or right in the Front viewport.

d) To make the glass smoother, select Glass_surface, click RevolutionGeometry1 in the Parameter Editor and set Subdivisions to 12.



Subdivisions 12

Angle 360.00

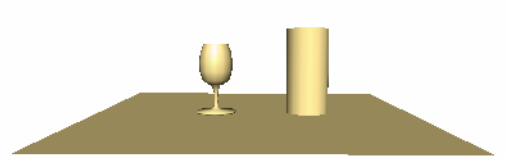
Top capping □

Your object is now smoother in longitude (look at it in the Top viewport). If you want a smoother object in latitude (Front and Right viewports), you must add points to Glass_curve.

e) To adjust the shape of the glass, select Glass_curve and edit it in Point mode. (Select the curve, press **4**, then select and translate individual points on the curve.)
The wine glass updates as you modify the original curve.

f) When the glass is to your liking, switch to Object mode (press **3**), select Glass_curve, and turn off its visibility.

g) Translate Glass_surface until it is next to the barrel.



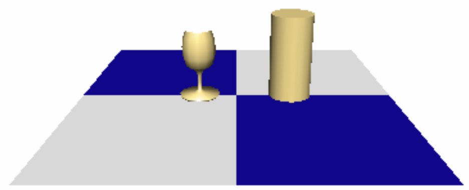
Adding materials

Use materials to adjust the color and physical properties of surfaces. Use textures to apply images to surfaces.

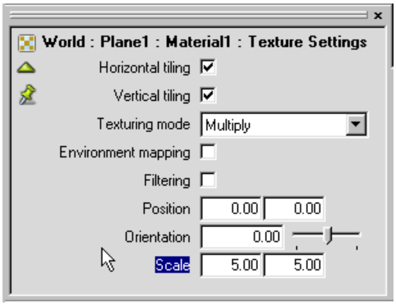
1. Add a checkerboard pattern to the table top:

   a) Select the plane, then choose Visualization > Add Texture.

   b) Navigate to the Tutorials / Textures folder and select the *Checkerboard.bmp* file.
   The checkerboard pattern is a 32 x 32 pixel image you apply to the tabletop like a single tile that covers the entire surface.



c) Now, make the checkers smaller by repeating the pattern. In the World Explorer, expand Plane / Material / Texture Settings and set the Scale to 5, 5.



World : Plane1 : Material1 : Texture Settings

Horizontal tiling ☑

Vertical tiling ☑

Texturing mode Multiply

Environment mapping □

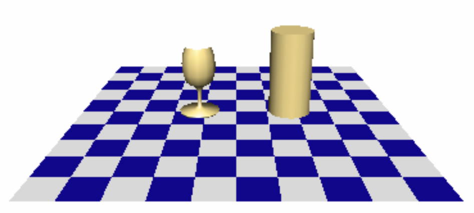
Filtering □

Position 0.00 0.00

Orientation 0.00

Scale 5.00 5.00

The checkerboard image is repeated five times in X and Y, making each checker five times smaller.

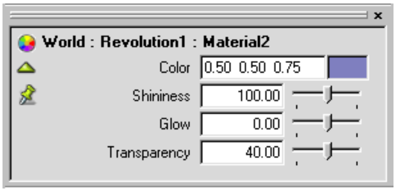


2. Make the glass transparent:

   a) Select Glass_surface.

   b) Choose Visualization > Add Material, then select a color.
   The default platinum color is replaced with a material that appears as Material1 under Glass_surface in the World Explorer.

   c) Expand Material1 and set the following parameters.



World : Revolution1 : Material2

Color 0.50 0.50 0.75

Shininess 100.00

Glow 0.00

Transparency 40.00

   d) Move the glass around the other objects to see the transparency effect.

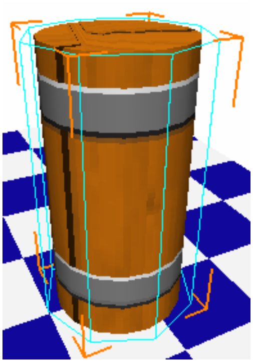
Note: Increasing the number of transparent objects in a World can slow down the refresh rate.

3. Add a wood panel texture to the barrel:

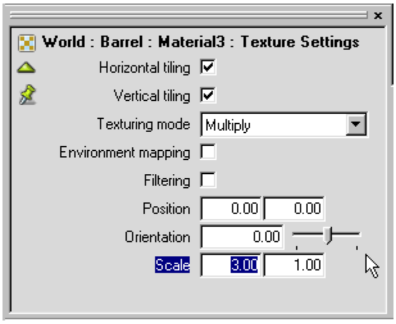
   a) Select the barrel and choose Visualization > Add Texture.

   b) Navigate to the Tutorials / Textures folder, and select *Wood_Cylinder.bmp*.
   The texture is applied with the default settings.

   c) Choose Visualization > Texture Projection > Cylindrical.
   The texture projector is added to the barrel. It can be manipulated like any other object.



   d) In the World Explorer, expand the barrel material and select the Texture Settings. Set the Scale to 3, 1 to create more horizontal repetitions.



World : Barrel : Material3 : Texture Settings

Horizontal tiling ☑

Vertical tiling ☑

Texturing mode Multiply

Environment mapping □

Filtering □

Position 0.00 0.00

Orientation 0.00

Scale 3.00 1.00

Adding a light to the world

Replace the default light with a spotlight to create a more dramatic effect.

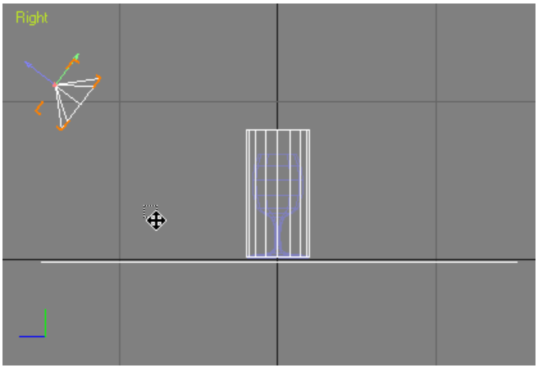
1. Choose Visualization > Add Light > Spot.
   A new light is added to the World.

2. To create a more dramatic effect, delete DirectionalLight1, the default light, from the World Explorer.

   The World goes dark.

3. Select the spotlight.

4. In the Right viewport, translate the light towards the upper-left corner and rotate it down to shine on the table.



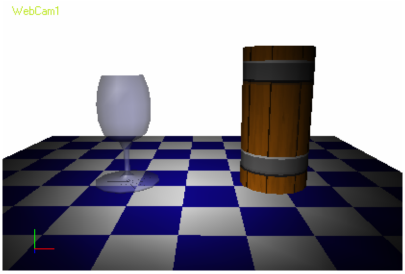
   Observe the changes in the WebCam viewport and manipulate the spotlight until you are satisfied with the results.

Animating the light and WebCam

Animate the WebCam and spotlight so that the WebCam zooms in and the spotlight turns on within the first 5 seconds of your opening a web page.

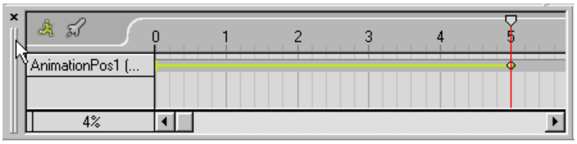
1. Animate the WebCam:

   a) Zoom and orbit in the WebCam viewport so that the view appears as it should at the end of the animation. Later on you will move the glass around, so zoom out far enough to obtain a good view of the entire table.



b) Stop the playback if applicable and go to time 5.

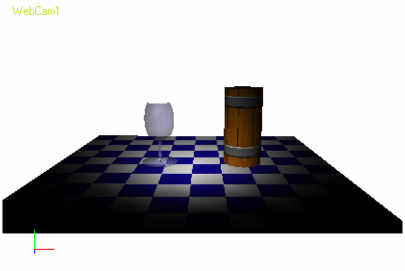
c) In the World Explorer, select WebCam, turn on the Translate button, and then choose Animation > Add Animation Sequence.
An animation sequence is created for the position of the WebCam. The animation sequence with one key appears both in the Sequencer and under the WebCam in the World Explorer.



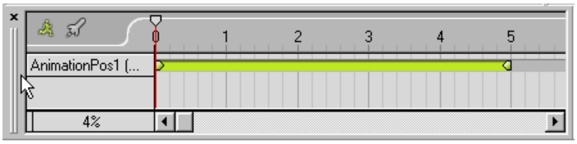
d) Go to the start of the animation at time 0.

e) Turn on the Record button.

f) Hold down **Z** and drag the cursor in the WebCam viewport to zoom out. Adjust the zoom to the point where you want the animation to start.



A second key appears in the Sequencer.



2. Switch to Browse mode to preview the result.

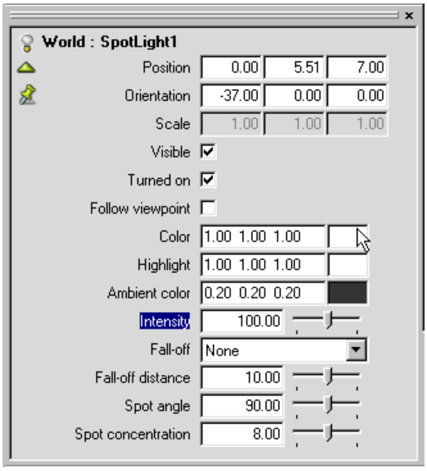
   The WebCam zooms in during the first 5 seconds before coming to rest at its final position.

3. Switch back to Author mode to continue working in the World.

4. Animate the light intensity so that the World starts out dark then gradually gets brighter as the WebCam zooms in:

   a) Go to time 5.

b) Select the spotlight. By default the intensity is set to 100, the maximum value.



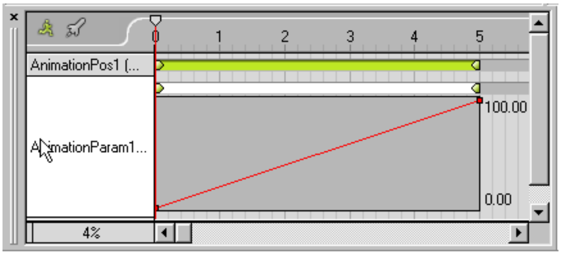
World : SpotLight1

Position	0.00	5.51	7.00
Orientation	-37.00	0.00	0.00
Scale	1.00	1.00	1.00
Visible	☑		
Turned on	☑		
Follow viewpoint	□		
Color	1.00 1.00 1.00		
Highlight	1.00 1.00 1.00		
Ambient color	0.20 0.20 0.20		
Intensity	100.00		
Fall-off	None		
Fall-off distance	10.00		
Spot angle	90.00		
Spot concentration	8.00		

c) In the World Explorer, select SpotLight / Intensity, then choose Animation > Add Animation Sequence.
A second animation sequence appears in the Sequencer.

d) Go to time 0.

e) With the light still selected, set Intensity to 0 in the Parameter Editor.
In the Sequencer, a key is added at time 0.



f) Turn on Play to preview the animation.

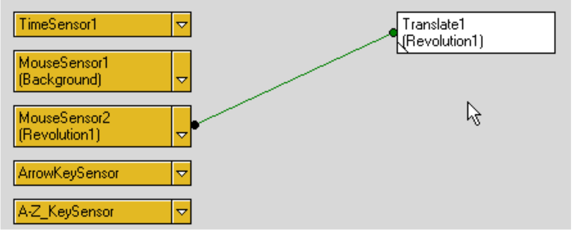
Adding interaction

Add a handle to the wine glass, and have it make a shattering sound when it gets within a specified distance from the barrel.

1. In the Top viewport, move the glass so that it is in line with the barrel.
   Align the glass and barrel to make it possible for them to collide when you drag the glass from left to right along the X axis.

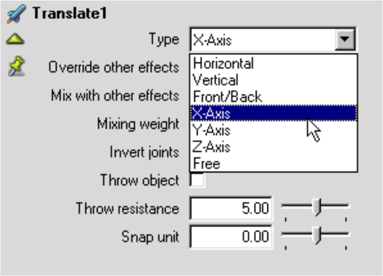
2. Add a handle to the glass:

   a) Select the glass and choose Interaction > Handles > Translate.
   In the Interaction Editor, the Translate1 reaction is linked to the Glass_surface mouse sensor.



   b) In the Interaction Editor, select the Translate1 reaction.
   The reaction's parameters appear in the Parameter Editor.

   c) Set the Type parameter to X-Axis so that the glass only moves along its X axis.



Translate1
Type: X-Axis
Override other effects
Mix with other effects
Mixing weight
Invert joints
Throw object □
Throw resistance: 5.00
Snap unit: 0.00

3. Create a reaction that plays a sound:

   a) Choose Interaction > Reactions > Play Sound.

   b) Navigate to Tutorials / Sounds, select *Smash.wav,* and click Open.

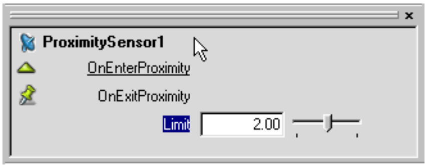
   c) A play sound reaction (PlaySound1) appears in the Interaction Editor.

4. Create a proximity sensor:

   a) Multi-select the glass and barrel, in that order.

b) Choose Interaction > Sensors > Proximity.
   A proximity sensor (ProximitySensor1) appears in the Interaction Editor.

c) Select the proximity sensor.
   Its parameters appear in the Parameter Editor.



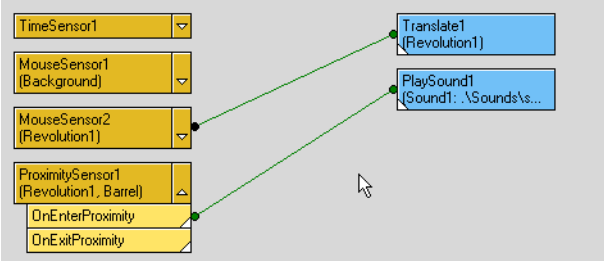
Limit 2.00

d) Set the Limit to 2.
   This causes the proximity sensor to trigger a reaction when the glass comes within a 2-unit radius of the barrel.

5. Link the proximity sensor to the sound reaction:

   a) In the Interaction Editor, expand the proximity sensor.

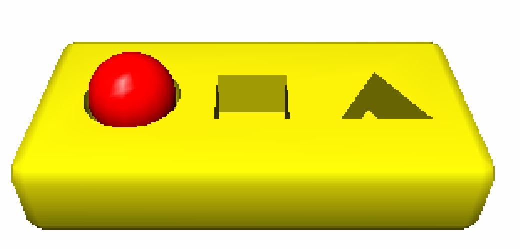
   b) Click and drag the cursor to draw a link from OnEnterProximity to PlaySound1.



6. Switch to Browse mode and test the interaction. Move the wine glass through the barrel to hear the crashing sound.

3 Relations and Interaction
IQ Test

In this lesson, you import VRML geometry created in a third-party modeling program. You add objects and interaction so that surfers can manipulate the objects on a web page. You create relations between the objects so they can only be moved within the limits that you set up.



Need Help? — Refer to the *IQ_test_complete.axel* file in the AXEL / Documentation / Tutorials folder.

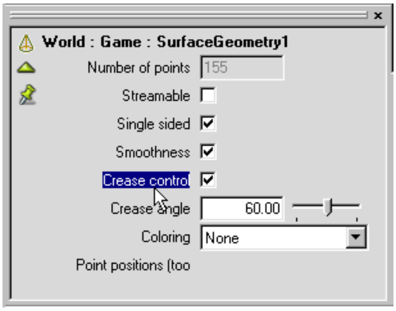
In this lesson:

Importing the game

An object was created in another 3D modeling application and then exported as VRML 97. Import the VRML file into AXEL so that you can add interaction and publish the content to the web.

Note: For more information on importing VRML, choose Help > OnLine Help, and look up Modeling / Importing Objects into AXEL.

1. Start a new project.
2. Import the VRML file:
   a) Choose File > Import.
   b) Navigate to Documentation/VRML, and select IQ_Game_WRL.
   The yellow game appears.
3. Since the game is probably too big for your viewports, you will have to zoom out and frame the game object in the viewports. To do so, press **SHIFT+A** or choose View > Viewpoint > Frame All.
4. Rename the object. In the World Explorer, select ChamferBox01-FACES, click twice, and type *Game*.
5. Improve the object's appearance:
   a) With the game still selected, click SurfaceGeometry1 in the Parameter Editor.
   b) Turn on Crease control.



World : Game : SurfaceGeometry1

Number of points 155

Streamable □

Single sided ☑

Smoothness ☑

Crease control ☑

Crease angle 60.00

Coloring None

Point positions (too

Notice how the object's appearance improves. When Crease control was off, AXEL was trying to smooth out all the object's corners, even corners that were not meant to be smooth, like the top edges of the holes. When Crease control is on, AXEL does not try to smooth out 90-degree angles.

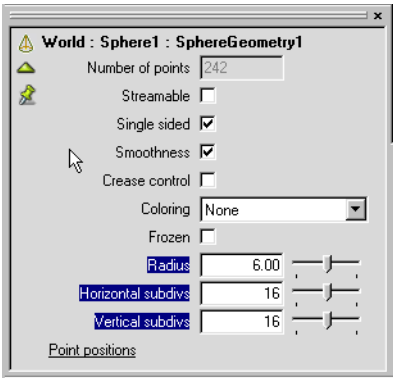
6. Turn Smoothness on and off to see its effect.

Note: For more information on using Crease control and Smoothness, choose Help > On-Line Help, and look up Modeling / Creating Surfaces / Surface Parameters.

Adding a sphere, a translate handle, and snap constraints

Create a sphere to fit in the round hole, then add a handle to the sphere so it can be manipulated on a web page. To help the surfer put the sphere in the right hole, use snap constraints; they will make the sphere stick to invisible objects that you place around the game.

1. Create a sphere:
   a) Choose Modeling > Preset Surfaces > Sphere.
   b) With the sphere selected, click SphereGeometry in the Parameter Editor.
   c) Set the following parameters.



World : Sphere1 : SphereGeometry1

Number of points 242

Streamable □

Single sided ☑

Smoothness ☑

Crease control □

Coloring None

Frozen □

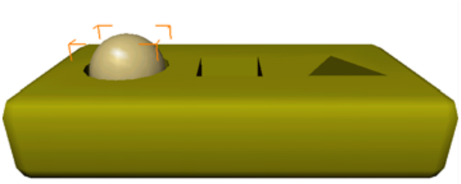
Radius 6.00

Horizontal subdivs 16

Vertical subdivs 16

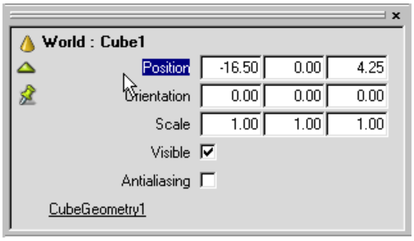
Point positions

   d) Move the sphere into the spherical hole.



2. Create three dummy objects to which the sphere can snap:
   a) Choose Modeling > Preset Surfaces > Cube.
   b) Duplicate it twice (press **CTRL+D**) to get three small cubes.

c) Rename the first cube *In_hole* and translate it so that it is in the center of the spherical hole.



World : Cube1

Position -16.50 0.00 4.25

Orientation 0.00 0.00 0.00

Scale 1.00 1.00 1.00

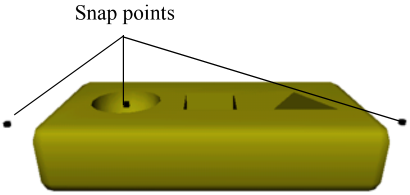
Visible ☑

Antialiasing ☐

CubeGeometry1

d) Select the second cube, rename it *To_left,* and translate it to the left of the game.

e) Select the third cube, rename it *To_right*, and translate it to the right of the game. Zoom (hold **Z** and drag) or pan (hold **X** and drag) to get a better view of the objects.



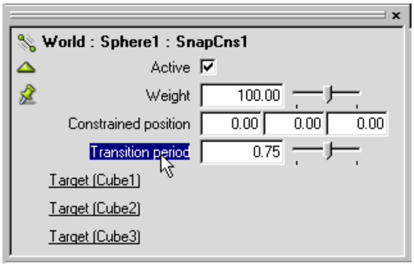
3. Add contraints to make the sphere snap to the cubes:

a) Select the sphere, then multi-select the three cubes.

b) Choose Relations > Snap. You can do this one cube at a time, or all cubes at once. The sphere moves to the closest cube.

c) Click an empty part of a viewport to deselect all objects, then select the sphere.

d) In the Parameter Editor, click SnapCns1 and set Transition period to 0.75.



World : Sphere1 : SnapCns1

Active ☑

Weight 100.00

Constrained position 0.00 0.00 0.00

Transition period 0.75

Target (Cube1)

Target (Cube2)

Target (Cube3)

When you drag the sphere away from one of the cubes, it will take 0.75 seconds to snap back to the closest cube.

Note: For more information on the snap constraint parameters, choose Help > On-Line Help, and look up Relations / Basic Constraints / Snap Constraint.

4. Preview the result:

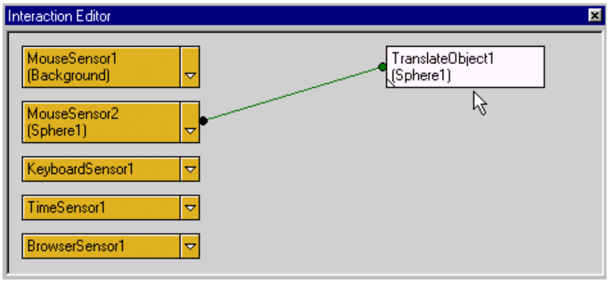
   a) Select the sphere and choose Interaction > Handles > Translate so you can move the sphere in the published content.

   b) Switch to Browse mode (press **CTRL+B**), and drag the sphere away from the game. It snaps to the closest cube.

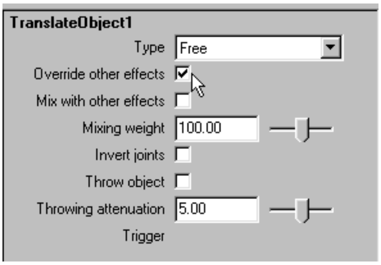
   c) Switch back to Author mode (press **CTRL+B**) to continue working in the World.

5. Adjust the snap constraint so that the sphere waits until you let go of the mouse button before it snaps to a cube:

   a) In the Interaction Editor, select the Translate1 (Sphere1) reaction.



   b) In the Parameter Editor, turn on Override other effects.



TranslateObject1

Type: Free

Override other effects ☑

Mix with other effects ☐

Mixing weight: 100.00

Invert joints ☐

Throw object ☐

Throwing attenuation: 5.00

Trigger

6. Move the sphere to the left so it starts on that side of the game.

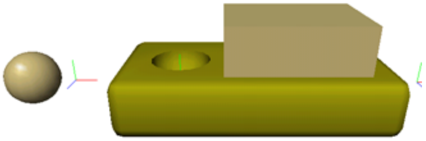
7. Select the cubes one at a time and turn off their visibility.

8. Switch to Browse mode and try the game.

Keeping the sphere outside a bounding box

To make it impossible to fail the IQ test, prevent the sphere from going anywhere near the square and triangular holes. To do this, create a cube around these holes and use a bounding box constraint to keep the sphere outside of that cube.

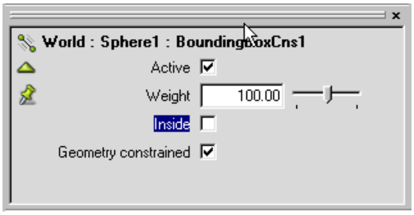
1. Create a cube:
   a) Switch to Author mode.
   b) Choose Modeling > Preset Surfaces > Cube.
   c) Scale the cube so it covers the square and triangular holes but not the spherical hole.



2. Choose View > Display > Constraints.

   Now constraints will appear in the viewports when you create them.

3. Constrain the sphere so it remains outside the cube:
   a) Multi-select the sphere and the cube, in that order.
   b) Choose Relations > Bounding Constraints > Box.
   The sphere moves inside the cube.
   c) In the World Explorer, expand the Sphere and select BoundingBoxCns1(Cube1).
   d) In the Parameter Editor, turn off Inside.



World : Sphere1 : BoundingBoxCns1

Active ☑

Weight 100.00

Inside □

Geometry constrained ☑

   Now the sphere moves out of the box; it can no longer go inside it.

4. Select the cube and turn off its visibility.

5. Switch to Browse mode and test the interaction.

Adding a webcam handle

Add a webcam handle so you can zoom in and out of the published content.

1. Add webcam handles:
   a) Choose Interaction > Handles > Orbit WebCam.

   b) Switch to Browse mode and test the interaction.

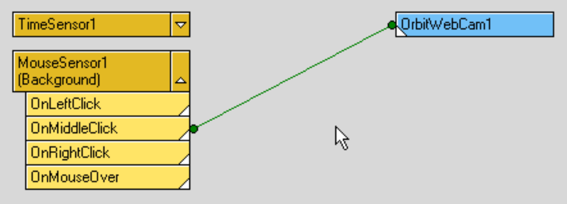
2. Change the webcam handle to use a different mouse button:

   a) Drag the Interaction Editor from its docked position and enlarge it.

   b) Expand the MouseSensor1(Background).
   OnLeftClick is linked to OrbitWebCam1.

   c) **SHIFT-**click OnLeftClick and ZoomCamera1 to multi-select them, then press **CTRL+U** to unlink them.

   d) Click and drag the cursor from OnMiddleClick to ZoomCamera1 to draw a link between the two. You can also multi-select them and press **CTRL+L**.



   e) Switch to Browse mode to test the interaction.

3. Publish the scene:

   a) Save an AXEL Stream.

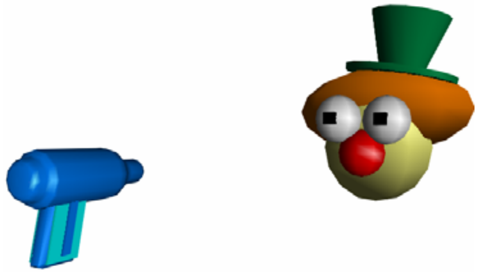
   b) Save a host web page using a 3D window size of 500 x 300 pixels.

   c) Open the host web page in a web browser and try the game.

4 Relations and Interaction

Clown and Pistol

You can create interesting interactions by setting up relations with the cursor. In this lesson, you work with a water pistol and a clown. You force the water pistol to follow the web cursor and the clown to look at the cursor. You also add a bounding box to ensure that the pistol does not move out of view.



Need Help? — Refer to the *Clown_pistol_complete.axel* file in the AXEL / Documentation / Tutorials folder.

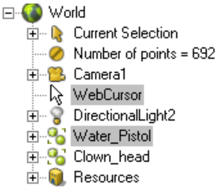
In this lesson:

Creating a relation with the surfer's cursor

1. Choose File > Open, navigate to *Clown_pistol_start.axel* in the Documentation / Tutorials folder, and click Open.
   There are two groups of objects in the World: the Water Pistol group and the Clown head group.

2. Make the water pistol follow the web cursor:

   a) In the World Explorer, multi-select the Water pistol group and the WebCursor object, in that order.



```
World
  Current Selection
  Number of points = 692
  Camera1
  WebCursor
  DirectionalLight2
  Water_Pistol
  Clown_head
  Resources
```

   b) Choose Relations > Pose Constraint.

   c) Switch to Browse mode.
   As you move the cursor, the water pistol follows. The pose constraint limits the water pistol's position to the position of the cursor.

   d) Switch to Author mode to continue working in the World.

Keeping the water pistol inside a bounding box

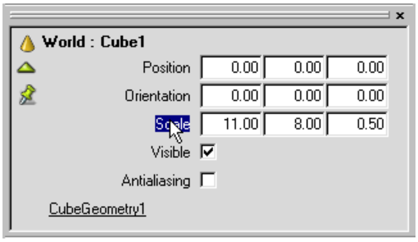
Create a bounding box constraint to ensure that the water pistol does not move out of sight.

1. When working with relations, it is useful to display constraints in the viewport. Choose View > Display > Constraints or press **D**.

2. Create a box:

   a) Choose Modeling > Preset Surfaces > Cube.

   b) Scale the cube to 11, 8, 0.5.

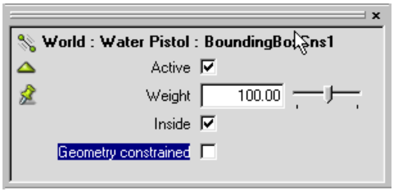


World : Cube1

Position	0.00	0.00	0.00
Orientation	0.00	0.00	0.00
Scale	11.00	8.00	0.50
Visible	☑		
Antialiasing	□		

CubeGeometry1

3. Constrain the water pistol to stay inside the box:
   a) Multi-select the Water Pistol group and the cube, in that order.
   b) Choose Relations > Bounding Constraints > Box.
      The water pistol is constrained to stay inside the box. By default, AXEL creates a bounding radius around the constrained object and forces it to stay within the bounding box (for example, making sure the whole chair does not go through the table). In this case we will only ask that the Water Pistol group's center stay within the bounding box.
   c) In the World Explorer, select the Water pistol group.
   d) Click on BoundingBoxCns1 in the Parameter Editor.
   e) Turn off Geometry constrained.



World : Water Pistol : BoundingBoxCns1

Active ☑

Weight 100.00

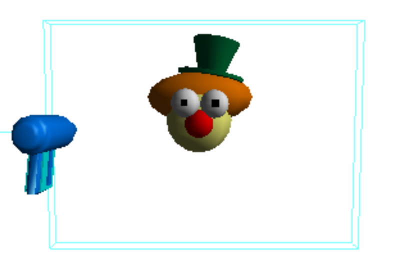
Inside ☑

Geometry constrained □

   f) Select the cube and turn off its visibility.

4. Switch to Browse mode to test the constraint.

   The water pistol follows the cursor but stays inside the bounding box.

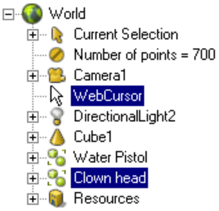


Making the clown point at the web cursor

Use a direction constraint to control the clown using the cursor.

1. Create a direction constraint between the clown and the cursor:
   a) Switch to Author mode.

b) In the World Explorer, multi-select the Clown head group and the WebCursor object, in that order.

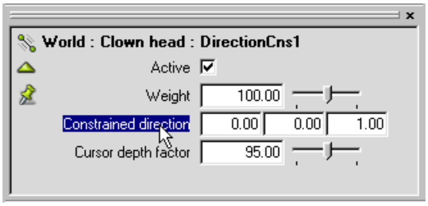


c) Choose Relations > Direction Constraint.

d) Switch to Browse mode to test the constraint.
The clown seems attached to the cursor by a string coming out of its left ear. To make the clown look at the cursor instead of listening to it, adjust the constraint parameters.

e) Select the Clown Head group, and click on DirectionConstraint to display its parameters.

f) Set Constrained direction to 0, 0, 1.



World : Clown head : DirectionCns1

Active ☑

Weight 100.00

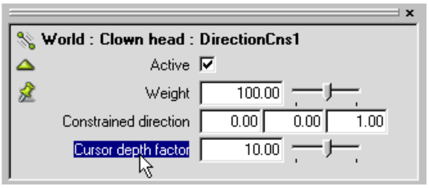
Constrained direction 0.00 0.00 1.00

Cursor depth factor 95.00

Now the axis coming out of the clown's nose (the local Z axis) is constrained to stay in the direction of the cursor.

g) Switch to Browse mode to test the constraint, then switch back to Author mode.

2. The Cursor depth factor indicates where the cursor is in relation to the camera. Adjust the Cursor depth factor:



World : Clown head : DirectionCns1

Active ☑

Weight 100.00

Constrained direction 0.00 0.00 1.00

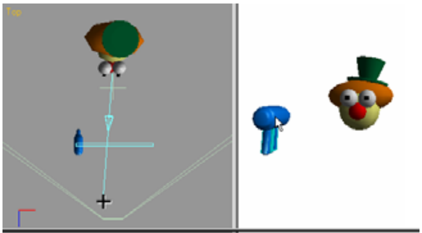
Cursor depth factor 10.00

a) By default, the Cursor depth factor is set to 95 %. Switch to Browse mode to test the interaction.
The cursor is far from the camera and close to the clown, so the clown has to turn sharply to follow the cursor.

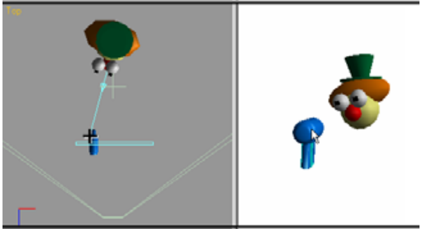
b) Switch to Author mode, and set the Cursor depth factor to 50%, then switch to Browse mode and test the interaction.
The cursor is halfway between the camera and the object; the clown can still follow the cursor without making sudden movements.

c) Switch to Author mode, and set the Cursor depth factor to 10 %, then switch to Browse mode and test the interaction.
In this case the cursor is at 10% of the distance between the camera and the object. The clown's head turns smoothly to follow the cursor. Notice that as the cursor moves away from the clown, he does not have to turn his head so suddenly to follow the cursor.

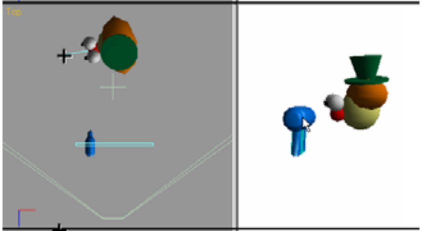
You can increase the Cursor depth factor above 100%, making the cursor move to the other side of the clown. To follow the cursor, the clown has to face away from the camera.



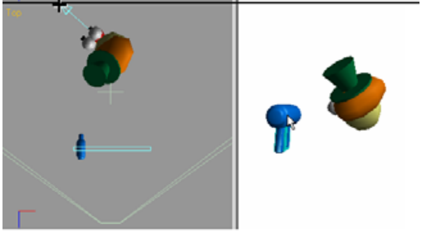
Cursor depth: 10%



Cursor depth: 50%



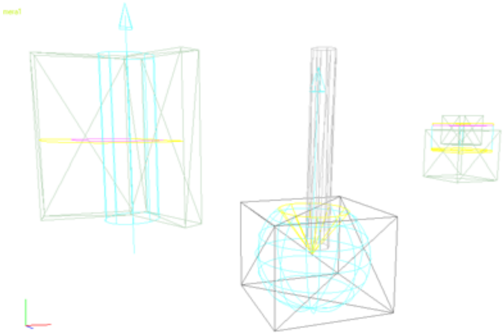
Cursor depth: 95%



Cursor depth: 150%

5 Relations and Animation
Pin, Ball, and Slider

In this lesson, you create three types of joint constraints: a pin joint, a ball joint, and a slider joint.



Need Help? — Refer to the following files in the AXEL / Documentation / Tutorials folder: *PinJoint_completed.axel*, *BallJoint_completed.axel*, and *SliderJoint_completed.axel*.

For more practice with joint constraints, see "Furniture" on page 77.

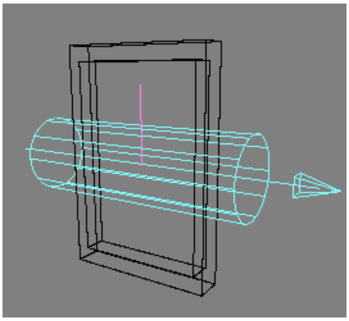
In this lesson:

Making a pin joint

Pin joints simulate the movement of a hinge and are ideal for creating effects like a door opening or a movie-clapper shutting.

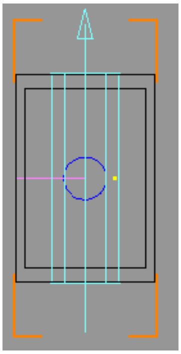
1. Make a door and a doorframe:
   a) Choose Modeling > Preset Surfaces > Cube.
   b) Scale it in X and Y until it resembles a door.
   c) Duplicate the door (**CTRL+D**) to make the doorframe. Scale the doorframe to make it bigger than the door.
2. In the World Explorer, rename the first cube *Door* and the second *DoorFrame*.
3. Add a constraint to the door:
   a) Multi-select Door and DoorFrame, in that order.
   b) Choose Relations > Joint Constraints > Pin.
   The door is repositioned on top of the doorframe.
4. Choose View > Display > Constraints.

   A blue cylinder with a pin through it appears in the viewports. This cylinder represents the pin joint.

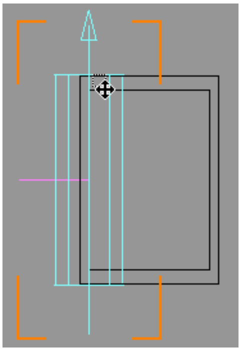


5. Adjust the constraint:

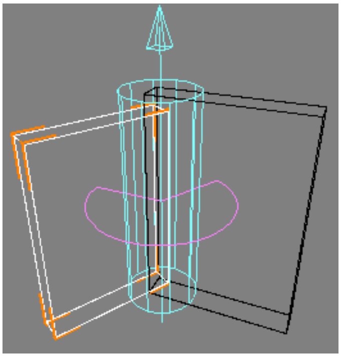
   a) Select and rotate the blue cylinder in the Front viewport so it stands vertically.



   b) Translate the blue cylinder to the left in the Front viewport so that it is at the left edge of the door, where the hinge should be.



6. Select the door and translate it.



It rotates around the pin joint as a door would rotate on a hinge.

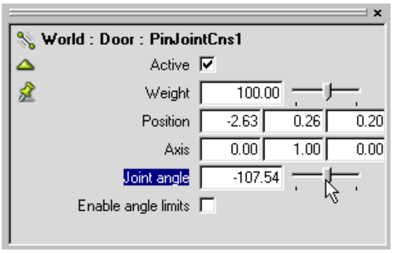
7. Translate the doorframe.

   The door follows accordingly.

8. Add some angle limits to the pin joint to make sure the door stops at the frame and does not swing in too widely in an arc:

   a) Select the blue pin joint cylinder.

   b) In the Parameter Editor, drag the Joint angle slider from left to right to experiment with door angles.



World : Door : PinJointCns1

Active ☑

Weight 100.00

Position -2.63 0.26 0.20

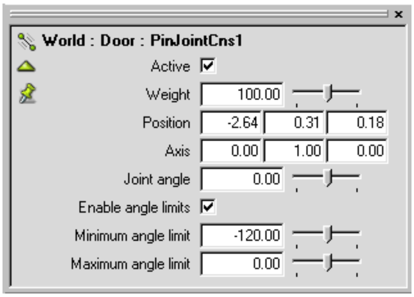
Axis 0.00 1.00 0.00

Joint angle -107.54

Enable angle limits ☐

   The Joint angle describes the position in degrees of the door around the frame in degrees. While translating the door is more intuitive, it is more effective to use the Joint angle to animate joints.

   c) Turn on Enable angle limits, and set the Minimum angle limit to -120.00 and the Maximum to 0.00.



World : Door : PinJointCns1

Active ☑

Weight 100.00

Position -2.64 0.31 0.18

Axis 0.00 1.00 0.00

Joint angle 0.00

Enable angle limits ☑

Minimum angle limit -120.00

Maximum angle limit 0.00

.

   d) Select the doorframe and choose Interaction > Handles > Translate.

   e) Repeat the previous step for the door.

9. Switch to Browse mode to test the interaction.

   When you move the doorframe, the door follows and swings back and forth since it has 120 degrees of freedom.

Animating the pin joint

Animate the pin joint angle so that the door opens automatically.

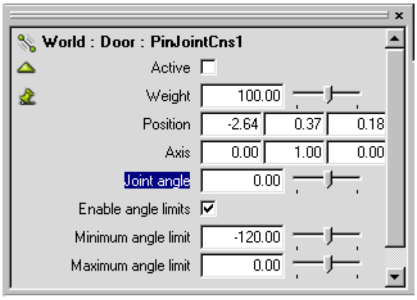
1. Create an animation sequence for the pin joint angle parameter:
   a) Turn off playback. Press **P** or turn off the Play button (▶).

   b) Go to time 0. Drag the playhead to time 0, type 0 in the Time field, or click the Go to Start button(⏮).

   c) Select the pin joint constraint by selecting the blue cylinder or by selecting the door and clicking on PinJointCns1 in the Parameter Editor.

   d) In the Parameter Editor, click the Pin icon(📌) to prevent the Parameter Editor from changing if you select another item.

   e) Select the Joint angle slider and drag it to 0.0.



World : Door : PinJointCns1

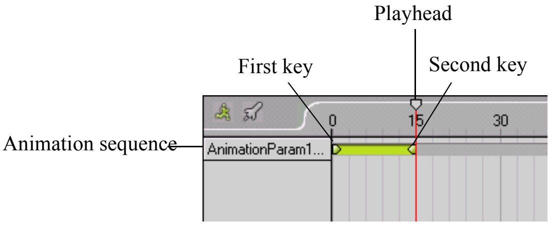
Active	□		
Weight	100.00		
Position	-2.64	0.37	0.18
Axis	0.00	1.00	0.00
Joint angle	0.00		
Enable angle limits	☑		
Minimum angle limit	-120.00		
Maximum angle limit	0.00		

2. Record two keys: one for the door closed and one for the door open:
   a) To record keys, turn on the Record button. (●)

   b) At time 0, make sure the Joint angle is set to 0.
   This is how the animation will start. It is the first key in the animation.

   c) Go to time 15, and set the Joint angle to -90.00.
   This is the next key in the animation.

   d) In the Sequencer, notice the animation sequence with the two keys.



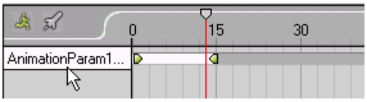
   e) To preview the animation, drag the playhead back and forth, or go to time 0 and press **P**.

   f) Switch to Browse mode and watch the door open.

Using a background sensor to trigger the animation

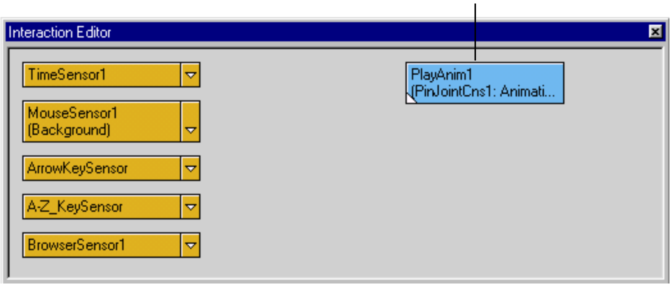
Currently the animation plays as soon as you switch to Browse mode. If you publish the content as it is, the door will open as soon as you open the web page. Instead, make the animation play only when you want it to. In this example, make the door open when the surfer clicks the background:

1. Convert the animation into a reaction:
   a) In the Sequencer, select the animation sequence.



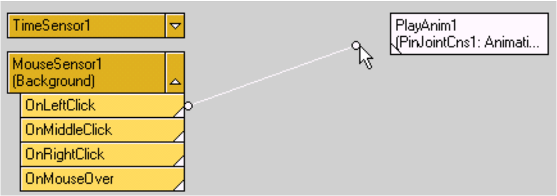
   b) Choose Interaction > Reactions > Play Animation.
   Notice the PlayAnim reaction in the Interaction Editor. You can resize or undock the Interaction Editor if you need more room to work on interactions.

Door opening animation



   Notice when you drag the playhead or turn on playback, the door no longer opens. When you converted the animation into a reaction, it was de-activated. Now it only plays when it is triggered.

2. Link a sensor to the reaction so that you can trigger it:
   a) Expand MouseSensor (Background), and click and drag from OnLeftClick to the PlayAnim reaction.

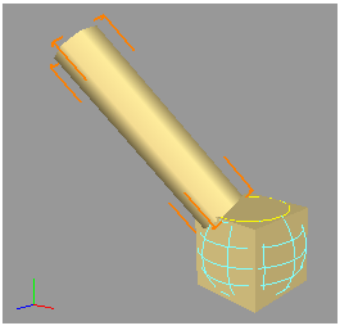


   b) Switch to Browse mode to test the interaction.

Now the animation will start when you click an empty part of the World (the background). Since the door opening is a reaction, when it is finished playing, the translate handle can take effect.

Making a ball joint

The ball joint constrains the position of an object to another, while allowing for a certain amount of directional translation. Constrain a cylinder to a cube and set limits so it behaves like a joystick.

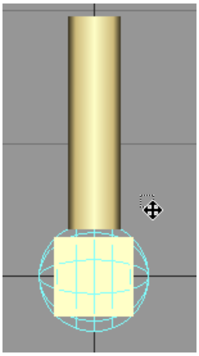


1. Create a cube, and scale it to 3, 3, 3. Rename it *Base.*
2. Create a cylinder and scale it to 1, 2, 1. Rename the cylinder *Joystick.*
3. Multi-select the joystick and the base, in that order.
4. Choose Relations > Joint Constraints > Ball.
5. Choose View > Display > Constraints.

   A blue sphere appears in the middle of the cube, representing the ball joint constraint.
6. Since the joystick is not correctly placed on the base, offset its position:

   a) Turn on the Translate button.

b) Holding the **U** key, drag the joystick where you like (while you do this, notice the Lock button on the main toolbar; it indicates that constraints are unlocked). When you are satisfied with the joystick's new position, release the **U** key.

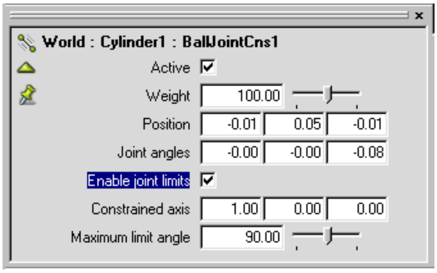


c) Translate the joystick. Notice that it is still too loose.
This is because the default setting of a ball joint is a full 360 degree freedom of rotation, while being constrained in position.

7. Set limits that govern joystick movement:

a) Select the blue sphere (the ball joint constraint).

b) In the Parameter Editor, turn on Enable joint limits.



World : Cylinder1 : BallJointCns1

Active ☑

Weight 100.00

Position -0.01 0.05 -0.01

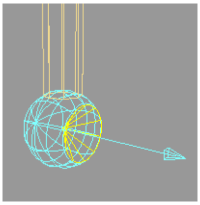
Joint angles -0.00 -0.00 -0.08

Enable joint limits ☑

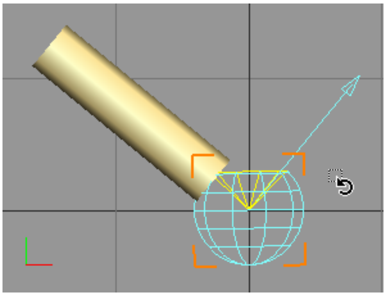
Constrained axis 1.00 0.00 0.00

Maximum limit angle 90.00

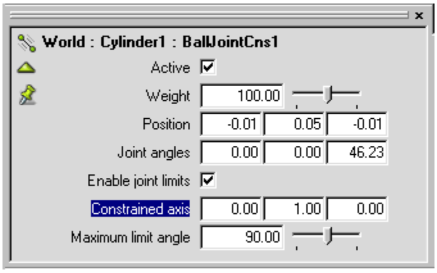
A yellow cone, representing the limits that restrict joystick movement, appears in the blue sphere.



c) Rotate the blue sphere so that the yellow cone is facing up. Notice the blue arrow pointing out of the yellow cone.



d) Change the constrained axis in the Parameter Editor to 0, 1, 0 to use the Y axis.



World : Cylinder1 : BallJointCns1

Active ☑

Weight 100.00

Position -0.01 0.05 -0.01

Joint angles 0.00 0.00 46.23

Enable joint limits ☑

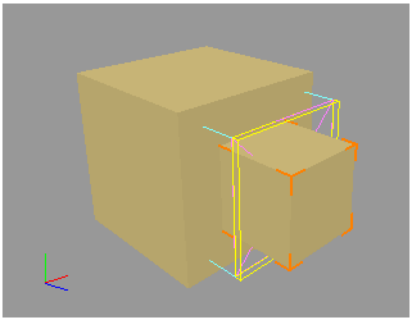
Constrained axis 0.00 1.00 0.00

Maximum limit angle 90.00

8. Select the constraint and set the Maximum angle limit to 120 in the Parameter Editor. (This creates a 120-degree aperture.)
9. Choose Interaction > Handles > Translate to apply translate handles on both the base and the joystick. Switch to Browse mode to test the interaction.

Making a slider joint

The slider joint constrains one object to another. It follows the constraining object but its movement is limited to one axis, much like a drawer that slides in and out of a dresser. In this exercise, make a smaller cube slide in and out of a larger one.

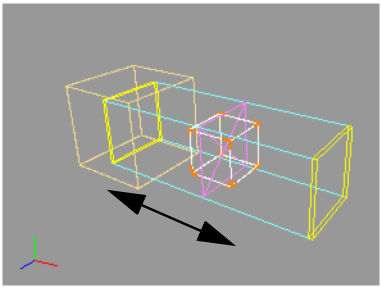


1. Choose Modeling > Preset Surfaces > Cube.
2. Duplicate the cube (**CTRL+D**), then scale it to twice its size. Rename it *large_cube*.
3. With large_cube selected, choose Modeling > Modify Surface > Rationalize Scale.

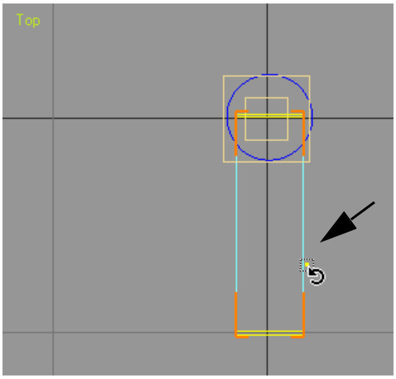
   The cube retains its size but the scaling values return to 1, 1, 1. This prevents odd offsets from occurring when the cube is manipulated.
4. Multi-select the small cube and large_cube, in that order. Choose Relations > Joint Constraints > Slider.
5. Check that View > Display > Constraints is on.

   Blue lines with yellow ends appear in the viewports. These lines indicate the axis along which the small cube will slide as well as its inner and outer limits of movement.

   For now, the cube can slide along its X axis.



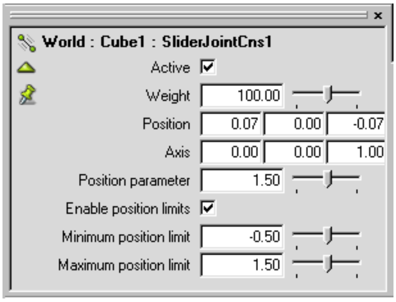
6. To make the cube slide toward the front along the Z axis, make sure the small cube is at position (0,0,0), then rotate the blue slider joint in the Top viewport.



7. Now, prevent the drawer from sliding out of the larger cube:

   a) With the blue slider joint selected, turn on the Enable position limits in the Parameter Editor.

   b) Modify the Minimum and Maximum position limit parameters. Set the minimum position limit to the back of the large cube, and set the maximum so that it is in front of the large cube.



World : Cube1 : SliderJointCns1

Active ☑

Weight 100.00

Position 0.07 0.00 -0.07

Axis 0.00 0.00 1.00

Position parameter 1.50

Enable position limits ☑

Minimum position limit -0.50

Maximum position limit 1.50

As you change the values, the slider joint updates in the viewports.

8. Select both cubes and choose Interaction > Handles > Translate to add handles to each cube.

9. Test the result in Browse mode.

6 Relations and Interaction
Furniture

In this lesson, you use joint constraints to add realism to a piece of furniture. Pin joints make the doors pivot on virtual hinges, and the bottom drawer opens and closes with a slider joint. A ball joint allows the lampshade to swivel around. These relations let you to constrain one object to another, while allowing a certain degree of freedom to the constrained object.

Need help? — Refer to the Furniture_Constraints_Completed.axel file in the AXEL / Documentation / Tutorials folder.

In this lesson:

Getting started

1. Open *Furniture_Constraints_start.axel* in the *Documentation / Tutorials folder*.

   The three objects that comprise the bottom drawer are grouped so they can be manipulated as one.

2. Since you are creating and manipulating constraints, display the constraints in the viewports:

   - To display constraints in all four viewports, choose View > Display > Constraints or press **SHIFT+D**.
   - To display constraints in one viewport, right-click the viewport and choose Display > Constraints or press the **D** hotkey.

     A few constraints already exist in the World. You may notice the constraints placed on the door handles, for example.

Making the doors swing on hinges

Constrain each door to a pin joint (the virtual hinge). The doors' knobs are already constrained, so all you have do is constrain Door_R and Door_L in the doors group.

1. Add a pin joint to the left door:
   a) In the World Explorer, expand the doors group.
   b) Multi-select Door_L and the table group, in that order.
   c) Turn on the Unlock button.
   d) Choose Relations > Joint Constraints > Pin.
   e) Turn off the Unlock button.
   A blue cylinder and PinJointCns1 are displayed if you expand Door_L in the World Explorer. By default, the door turns on the X axis and halfway between both object centers.

2. Position the constraint:
   a) Select the pin joint. In the Front viewport, rotate it into vertical position.
      As you do so, the Parameter Editor displays the pin joint constraint (PinJointCns). You can also type 0, 1, 0 in the axis box to specify the Y axis.

      The door rotates properly on its Y axis, but you must now position the hinge.

About unlocking constraints

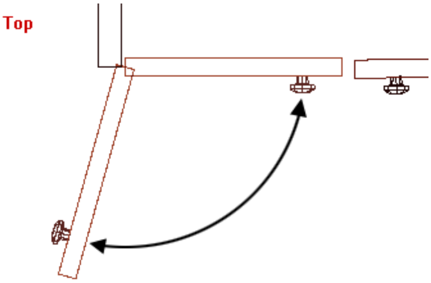
The Unlock button lets you create an offset when you initially set a constraint between objects. With no offset, the door would by default have been positioned over the table group's center. Using Unlock, you can also modify the placement of the constrained object at a later time, since it temporarily deactivates the constraint. It is therefore important to turn Unlock off once you are finished or satisfied with the location of the two objects.

b) Turn on the Translate button. In the Top viewport, move the pin constraint between the door and the side of the table. The position of your pin joint should be approximately –4.9, –2.0, 5.5. (The pin joint's actual position in the Y axis is not important.)

3. Set limits to the door movement:

   a) Select Door_L and turn on the Translate button.
   Move the door on its hinge.

   b) Select the pin joint. In the Parameter Editor, turn on Enable angle limits, and type 240 for minimum and 0 for maximum.
   This limits the door's freedom of movement so it stops in a closed position and does not go past the outer panel of the table.



4. Add interaction:

   a) Select Door_L and choose Interaction > Handles > Translate.

   b) Choose Interaction > Handles > Orbit WebCam.

5. Switch to Browse mode and try moving the left door. Click and drag in the background to orbit around the World.

6. Repeat steps 1-5 for Door_R.

Animating a door

Create an animation of the door opening and closing, then turn it into a reaction.

1. Create a position animation sequence:

   a) Turn off the Play button if time is running and return to time 0.

   b) Select Door_R, turn on the Translate button, and then turn on the Record button.
   This creates an animation, which appears in the Sequencer as AnimationPos.

2. Create keys:

   a) Go to time 1 and translate the door to an open position. (The Record button must be turned on for you to move the door.)

b) Go to time 2 and translate the door to a closed position.

3. Test and tweak the animation:
   a) Go to time 0 and turn on Play.
   The door swings open and closed.
   b) Move the keys on the Sequencer to adjust the timing.

Making a reaction out of door animation

If you switch to Browse mode, you see that the animation plays immediately. Now make the door's movement a reaction that you trigger yourself.

1. Create a reaction:
   a) In the World Explorer, expand Door_R and rename AnimationPos *open_and_close* (click twice on AnimationPos to modify its name).
   b) With the animation selected, choose Interaction > Reactions > Play Animation.
   Notice how the animation's active mode toggles off.

   The animation becomes a reaction, visible in the Interaction Editor and in the World Explorer under Resources / Reactions / PlayAnim1. It is now a reaction waiting for a sensor to trigger it.
2. Create a sensor to trigger the new reaction:

You could choose anything to trigger the reaction, such as a time sensor opening the door at 10 seconds or a button click from a remote control. In this case, use the door itself as the sensor.

a) Select Door_R and choose Interaction > Sensors > Mouse.

b) In the Interaction Editor, expand the mouse sensor.

c) Multi-select the OnLeftClick trigger and the PlayAnim1 reaction, and choose Interaction > Link Reaction.

3. Switch to Browse mode and left-click the door to trigger the animation.

Making the drawer slide

The drawer must slide in and out of the cabinet on a single axis, without falling out or going through the back panel. A slider joint is the perfect tool for this kind of motion.

The three elements that make up the drawer have been grouped. Apply the relation to the entire group.

1. Add a slider joint:
   a) In the World Explorer, multi-select the drawer group and the table group, in that order.

   b) Turn on the Unlock button.
   c) Choose Relations > Joint Constraints > Slider.
   d) Turn off the Unlock button and make sure the constraints are displayed (press **D**).
   e) Select the drawer group and turn on the Translate button. Notice how it only slides from left to right, along the axis. Next, correct this problem.

2. Position the constraint:
   a) Expand the drawer group in the World Explorer and select SliderJointCns.
   b) In the Top viewport, rotate SliderJointCns 90 degrees clockwise. In the Parameter Editor, check that the axis values are 0, 0, 1.
   c) Select the drawer group and translate it.
   It now slides back and forth properly.

3. Adjust the drawer's limits so it can slide out a little further:
   a) Select SliderJointCns.
   b) In the Parameter Editor, change its maximum position to 9.
   The drawer can now slide almost completely out of the cabinet.

4. Add interaction:
   a) Select the drawer group in the World Explorer and choose Interaction > Handles > Translate.
   b) Switch to Browse mode to test the interaction.

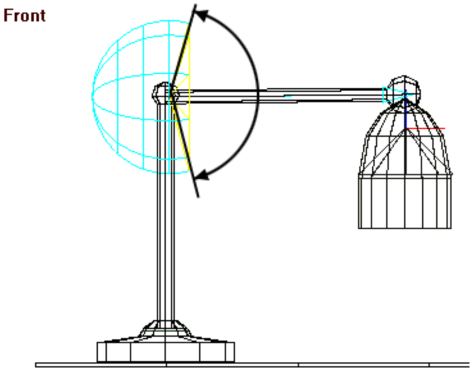
Using ball joints on the lamp

For the tabletop lamp, use ball joint constraints (universal joints) to manipulate the top tube and the lampshade. A ball joint allows full rotation of an object while constrained to another object's position.

About lighting

A spotlight has been pose-constrained to the lampshade, so when you manipulate the lampshade, you will light the tabletop surface. The tabletop's geometry, originally quite simple, has been subdivided (Modeling > Modify Surface > Subdivide). Subdivision was necessary because a surface with more detail is better able to represent lighting effects. If a surface is comprised of one triangle, it can only be lit or unlit. A more complex surface can display varied light intensity and falloff.

1. In Author mode, add a ball joint to the lamp arms:
   a) In the World Explorer, multi-select lamp_top_arm and lamp_bottom_arm (the order is always important since it determines what is constrained to what).
   b) Turn on the Unlock button.
   c) Choose Relations > Joint Constraints > Ball.
   With constraints displayed, a blue BallJointCns sphere appears (it also appears in lamp / lamp_top_arm).
   d) Turn off the Unlock button.
2. Position the ball joint:
   a) Translate BallJointCns so its center is positioned where both lamp components meet (position: -2.75, 5.0, –2.2).
   b) Translate lamp_top_arm in all four viewports.
   The joint now has a full 360-degree range of movement. Now set limits to that movement.
3. Limit the freedom of the top arm:
   a) Select BallJointCns and turn on Enable joint limits.
   b) Type 0, 1, 0 (the Y axis) for the constrained axis and use the slider to expand the maximum limit angle to 160.00.
   c) Now select lamp_top_arm and translate it in all four viewports.
   The arm moves freely within a 160-degree limit.



4. Add a ball joint for the lampshade:
   a) Move lamp_top_arm back into horizontal position so it touches the lampshade (see illustration above).
   b) In the World Explorer, multi-select LampShade and then lamp_top_arm, in that order.
   c) Turn on the Unlock button.
   d) Choose Relations > Joint Constraints > Ball.

A BallJointCns sphere appears, halfway between both object centers.

e) Turn off the Unlock button.

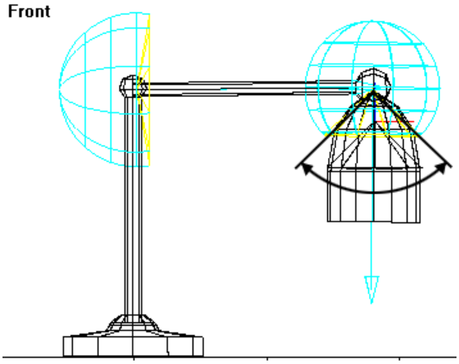
5. Position the ball joint:

   Translate the ball joint under LampShade so its center is where LampShade and lamp_top_arm meet (position: 2.25, 5.0, –2.2).

6. Limit the freedom of the lampshade's movement:

   a) With the ball joint still selected, turn on Enable joint limits and type 0, 1, 0 for the constrained axis.
   If you select the lampshade and translate it in all four viewports you will see how it moves freely within the permitted 90-degree range. In this case, the lampshade is pointing horizontally at a range of 90 degrees.

   b) To correct the lampshade's position, select the ball joint and rotate it 90 degrees clockwise in the Front viewport.



7. Add interaction:

   a) Select LampShade and choose Interaction > Handles > Translate.

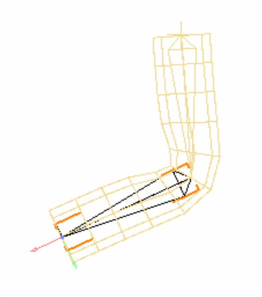
   b) Select lamp_top_arm and choose Interaction > Handles > Translate.

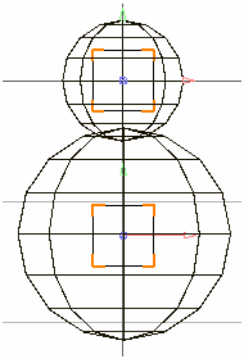
8. Switch to Browse mode and try out your various interactions.

7 Character Modeling with Limbs and Skin

Snowman

In AXEL, you use limbs to easily set up a skeletal structure and animate it. When you manipulate a limb, AXEL calculates the secondary movement of individual bones. Skin allows for an object to be deformed by limbs or any other object.





Need help? — Refer to the *Snowman_complete.axel* file in the AXEL / Documentation / Tutorials folder.

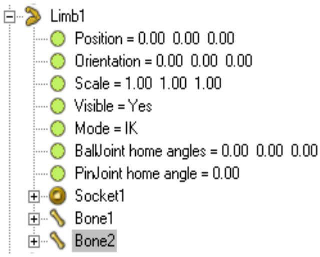
In this lesson:

Creating limbs

You can define 2-bone limbs that bend in all directions. If you want to change the bones' size, you can scale them.

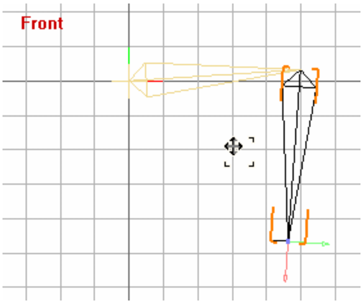
1. Create a 2-bone limb:
   a) Choose Modeling > Limbs > 2 Bones.
   A limb appears in the viewports.

   In the World Explorer, the limb appears as a group containing a socket and two bones, with the second bone selected.

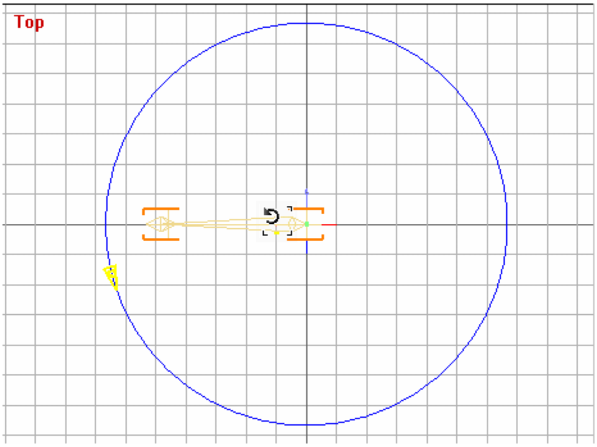


```
Limb1
  Position = 0.00  0.00  0.00
  Orientation = 0.00  0.00  0.00
  Scale = 1.00  1.00  1.00
  Visible = Yes
  Mode = IK
  BallJoint home angles = 0.00  0.00  0.00
  PinJoint home angle = 0.00
  Socket1
  Bone1
  Bone2
```

   b) Translate the second bone and notice how the bones are connected. When you move the second bone, the orientation of the first bone changes. By default, the limb can only bend inwards, like a knee or an arm.

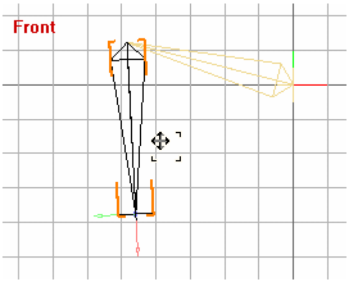


2. As it is, the knee can kick to the right. To make the knee point the other way and kick to the left, rotate the entire limb. In the World Explorer, select the limb group, and then rotate it in the Top viewport.



3. Select the second bone and translate it.

   It now bends in the opposite direction, so it can kick to the left.



4. Place the limb in a vertical position.

Creating skin

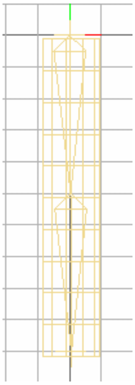
Now that you have a skeleton, you can make a skin to cover the bones. The skin can be a single surface or a group. For example, if you have a character composed of a head and a torso, you should group them, then make the group a skin over the skeleton.

1. Make a cylinder to use as skin:
   a) Choose Modeling > Preset Surfaces > Cylinder.

   b) In the World Explorer, expand the cylinder geometry.

c) In the Parameter Editor, set Height to 10.00 and Vertical subdivisions to 10. The additional subdivisions give the cylinder places to bend.

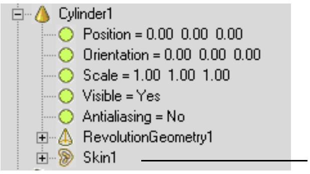
d) Place the cylinder over the limbs.



2. Skin the cylinder to the limb:

a) In the World Explorer, multi-select the cylinder and Limb1, in that order.

b) Choose Modeling > Skin.
The cylinder is now a skin on the limb. Expand the cylinder in the World Explorer to see the Skin deformer.

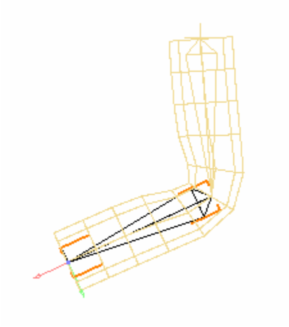


```
⊟ Cylinder1
    Position = 0.00  0.00  0.00
    Orientation = 0.00  0.00  0.00
    Scale = 1.00  1.00  1.00
    Visible = Yes
    Antialiasing = No
  ⊞ RevolutionGeometry1
  ⊞ Skin1
```

Skin deformer

3. Select the second bone and translate it.

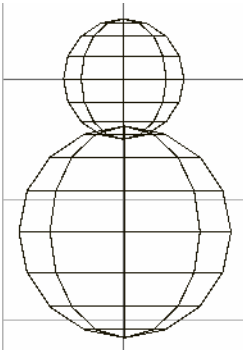
   The cylinder bends, following the movements of the bones.



Creating deformers and skin

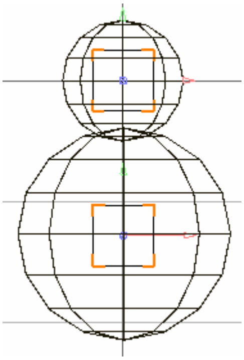
You do not need limbs to support a skin. You can use any object. The objects inside the skin are called deformers. When you move a deformer, the skin follows. In the following example, you use two cubes as deformers and a group of spheres as skin.

1. Create two spheres and group them:
   a) Choose Modeling > Preset Surfaces > Sphere.
   b) Duplicate the sphere, then scale the duplicate to twice its size.
   c) Translate it down to make a snowman.
   d) Multi-select both spheres and choose Relations > Group.



2. Place two cubes in the snowman and use them as deformers:
   a) Choose Modeling > Preset Surfaces > Cube. Duplicate the cube.

b) Place one cube in the snowman's head and one in his belly.

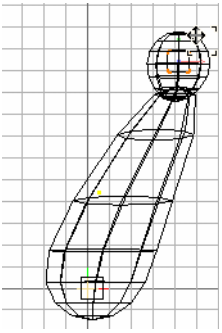


3. Make the spheres into skin over the cubes:

   a) In the World Explorer, multi-select the group of spheres and then the cubes, in that order.

   b) Choose Modeling > Skin.
   A skin deformer is added to each sphere in the World Explorer.

   c) Translate the top cube.
   The skin deformer makes the vertices of the spheres follow the cubes. This is fine if you choose to animate the cubes.



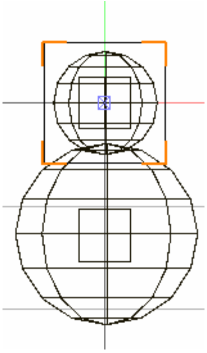
4. Allow the surfer to interact with the snowman.

   The cubes are what deform the snowman. The surfer cannot interact with them since they are hidden inside the spheres. Create a dummy object, make it transparent, and put a translate handle on it.

   a) Create a cube.

   b) Rename it *Dummy*.

c) Scale it so that it is larger than the snowman’s head, and translate it over his head.



d) Multi-select the top cube and the Dummy cube, in that order.

e) Turn on the Unlock button so you can create an offset.

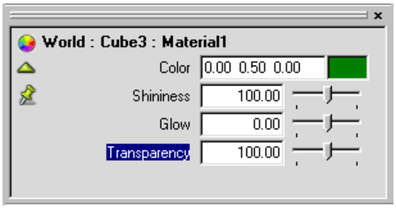
f) Choose Relations > Pose Constraint.

g) Turn off the Unlock button.
The top cube should now follow the Dummy cube around.

h) Select the Dummy cube and choose Visualization > Add Material.

i) Choose any color.

j) In the World Explorer, select the material. In the Parameter Editor, set a Transparency of 100.



World : Cube3 : Material1

Color 0.00 0.50 0.00 🟩

Shininess 100.00

Glow 0.00

Transparency 100.00

The Dummy cube should now be invisible.

k) In the World Explorer, select the Dummy cube and choose Interaction > Handles > Translate.

l) Switch to Browse mode. As the cursor moves over the transparent Dummy cube, the cursor switches to a hand and you can drag the head around.

8 Character Modeling and Animation

Mr. Cactus

In this lesson, you create and animate a simple character. To do this, you first create a skeleton with limbs and then create surfaces to use as skin. You use materials and textures to make the character come to life.

Need help? — Refer to the *Cactus_complete.axel* file in the AXEL / Documentation / Tutorials folder.

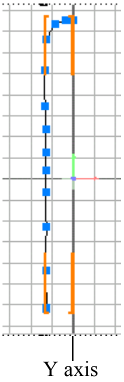
In this lesson:

Creating the cactus body from a revolution

Create the cactus body from a revolution.

1. Draw the profile curve of the cactus body:
   a) Choose Modeling > Draw Curve.

   b) In the Front viewport, click to draw points that form a curve. Use the Y axis as a reference since the curve will revolve around it.



   c) Press **ESC** twice to exit Draw Curve mode.

   Note: Pressing **ESC** once ends the curve without closing it and without exiting Draw mode, allowing you to draw another curve.

   d) To add points to the curve, choose Modeling > Modify Curve > Add Point, then click the curve in the viewport to insert new points.

   e) To move points on the curve, switch to Point mode (press **4**), turn on the Translate button, and click and drag the points you want to move.

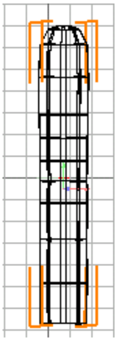
   f) To delete points, select them and press **DELETE**.

Note: Be sure to add enough points midway up the curve so that it will bend correctly.

2. Create the cactus body from the curve using revolution:
   a) Switch to Object mode (press **3**) and select the curve.

   b) Choose Modeling > Create Surface > Revolution from Curve.

c) In the Parameter Editor, click RevolutionGeometry to display the revolution parameters. Change Subdivisions to 16 to add points to the surface.



d) You can modify the surface by manipulating the curve. Select the curve, switch to Point mode, and translate points. The revolution surface updates as you change the curve. Arrange the top points so that the head is not pointy.

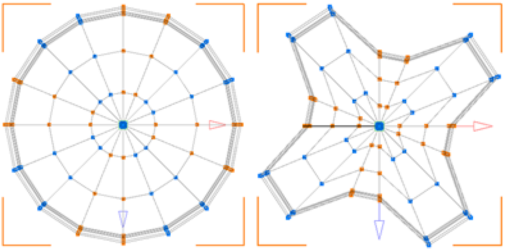
e) Rename the revolution surface *Cactus body.*

3. Modify the cactus body by manipulating its points:

a) Select Cactus body and frame it in Top view (press **F**).

b) Switch to Point mode (press **4**) and select 2 of every 4 rows. **SHIFT**-select to add points to the selection.

c) To scale the points, hold the **E** key, then click and drag the cursor in Top view. Look at the cactus in all four viewports, orbit in Webcam view, and adjust the shape if needed.

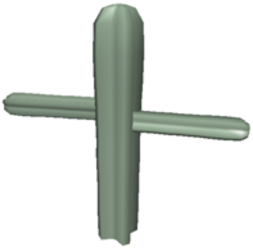


d) Manipulate the curve and notice that it is no longer linked to the cactus body. This is because the surface was automatically frozen when you edited points. You can restore the link to the curve by turning off the frozen parameter, but then the scaled points will return to their original shape.

e) Turn off the curve's visibility.

Duplicating the body to create arms

To create the arms, duplicate the cactus body and scale it into shape.



1. Make the first arm from a duplicate of the body:
   a) Press 3 to switch to Object mode, select the body, and choose Edit > Duplicate or press **CTRL+D**.

   The duplicate object is offset by 0.2 units on all three axes so that it stands out.

   b) Rename the duplicate *Arm_Right*.

   c) In Right view, rotate it 90 degrees counter-clockwise. Its Orientation value should be 90, 0, 0 in the Parameter Editor.

   d) Scale the arm to half its original size and translate it so that it is roughly halfway up the body.

2. Make the next arm from a duplicate of the first:

   a) Duplicate Arm_Right and rename it *Arm_Left.*

   b) In Right view, rotate the right arm 180 degrees and translate it into position on the other side of the cactus body.

3. To group the elements, select the body and both arms, and choose Relations > Group or press **CTRL+G**. Rename the group *My Cactus*.

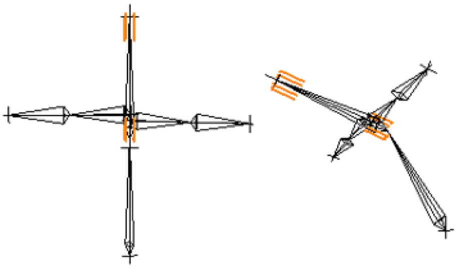
Controlling polygon weight

When creating content for the web, you should keep track of how detailed the geometry is that you add to your World. Even though downloading geometry is very efficient, each additional point affects download time for surfers. This is also true for textures and sounds.

To see how long it will take to download a World, choose Web Integration > Download Time Estimates.

Adding bones to the cactus

You can use inverse kinematics to manipulate the cactus. With inverse kinematics, you create limbs, make sure they fold correctly, and assemble them so they work together; then, when you manipulate a single bone, AXEL calculates the motion of all the other bones. For example, if you manipulate an ankle, the shin and thigh follow automatically.



1. Create the backbone:
   a) Choose Modeling > Limbs > 2 bones.
   A 2-bone limb appears in the viewports. In the World Explorer, notice that the limb is a group containing two 5-unit bones.
   b) Manipulate the backbone. To move the whole limb, select the limb group in the World Explorer and translate it. To bend the limb, translate bone 2. When you drag bone 2 in Front view, AXEL uses inverse kinematics to calculate the rotation of both bones. To bend the bone in another direction, rotate the limb group, then translate bone 2.
2. Move and scale the bones to adapt them to the cactus:
   a) Select the end bone and translate it so that the limb is in an upright position.
   b) Select the limb group in the World Explorer and translate it so that it is roughly at the base of the cactus body.
   c) To stretch the limb to the size of the cactus, scale each bone. Select the lower bone and scale it halfway up and then select the end bone and do the same.
   d) Rename the Limb group *Backbone*.
3. Add bones to the arms. The left arm should point up and the right arm should bend down.

   Note: With inverse kinematics, a 2-bone limb is like a leg. The knee only allows you to bend your shin in. You cannot twist your shin towards the right or left at the knee.

   a) Choose Modeling > Limbs > 2 bones.
   By default the 2-bone limb bends clockwise in Front view.
   b) Select the limb group in the World Explorer and rotate it 90 degrees counter-clockwise in Top view so that it is to the right of the cactus in Right view. Its Orientation should be 0, 90, 0.

c) Translate it so that it is inside the left arm of the cactus.

d) Rename the limb *left_arm_limb*.

e) Scale down the individual bones so that they fit inside the arm. If you select the end bone and translate it, you will see that it can only fold downwards. Notice the cactus skin does not follow. You will apply the skin soon.

4. Create the right arm:

   a) Choose Modeling > Limbs > 2 bones.

   b) Select the limb group in the World Explorer and rotate it clockwise 90 degrees in Top view so that it is to the left of the cactus in Right view.

   c) Translate it so that it is inside the right arm of the cactus.

   d) Rename the limb *right_arm_limb* in the World Explorer.

   e) Scale down the individual bones so that they fit inside the arm.
   Notice when you translate the end bone, it only bends downwards.

   f) To change the way the bones bend, rotate the limb group. Select the limb group in the World Explorer and rotate it 180 degrees in Front view.
   Now when you translate the end bone, it bends upwards. The right arm limb's Orientation should be 0, 90, 0.

5. Hide the cactus body and arms by turning off its group visibility.

6. Use a pose constraint to attach the arm to the upper body:

   Just as all the bones in a human skeleton are interconnected, you must assemble the various limbs in your character. Think about how your body is put together. When you bend forwards, the movement of your spine causes your head and shoulders to follow, but it does not affect your feet. To make sure the cactus's arms follow when you move the head, you constrain the arms to the upper backbone.

   a) Select the right_arm_limb group and then **SHIFT**-select the upper backbone.

   b) Turn on the Unlock button.

   c) Choose Relations > Pose Constraint.

   d) Repeat the same steps for left_arm_limb.

   e) Set the Unlock button to Locked.

7. Test the limbs by manipulating the end bones. Select the three end bones and translate them to get a better understanding of how they react.

8. Make the cactus body visible, then place the body in an upright position and arms outstretched so that it matches the cactus skin.

9. Reset the Home State of the limbs:

   a) Multi-select the three limbs and choose Modeling > Modify Limbs > Reset Home State.

The current angles and bending behavior become the Home State. You can come back to this state at anytime by choosing Modeling > Modify Limbs > Go to Home State.

Skinning geometry to limbs

Now that you have a skeleton, you can use the cactus as a skin to place over the limbs. The geometry of the cactus body and arms will be deformed by the limbs underneath it.

In some cases you can skin a group of objects to a group of deformers, especially if you want one limb to deform more than one object. In this example, skin the body and the arms individually to the underlying limb.

1. Skin the cactus body to the backbone:
   a) Make sure the backbone is in a straightened position and under the cactus body; otherwise, you are applying an offset in how they are linked.
   a) Select the cactus body and **SHIFT**-select the backbone limb group.
   b) Choose Modeling > Skin.
   If you now select the upper backbone and translate it, the cactus body is deformed by the underlying limb.
2. Repeat the previous step by skinning both arms to their respective limbs.
3. In the World Explorer, multi-select all three limb groups and group them.

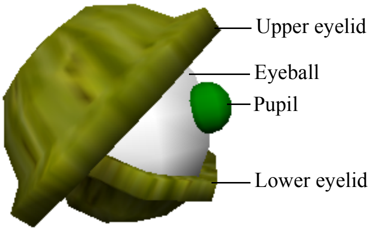
Adding thorns using instances

To decorate the cactus, add some thorns.

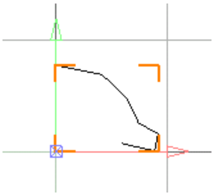
1. Create thorns:
   a) Choose Modeling > Preset Surfaces > Cone.
   b) In the Parameter Editor, click the cone geometry and change the width to 1 to have a pointier thorn.
   c) Scale it to an acceptable size.
   d) Create five to six instances. Choose Edit > Instance or press **CTRL+I**.
2. Translate, rotate, and scale the thorns into shape and position.
3. Pose constrain them individually to the nearest bone so that they follow the movements of the skeleton. (Remember to unlock constraints before applying them.)

Creating one eye from preset surfaces and revolutions

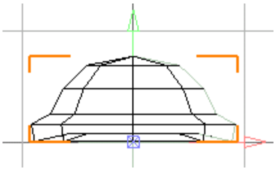
Create an eye using preset surfaces and revolutions.



1. To make room to work, turn off the visibility of the limb group and the cactus group.
2. Create an eyeball with a sphere:
   a) Choose Modeling > Preset Surfaces > Sphere.
   b) In the Parameter Editor, click the sphere geometry and set the radius to 0.60.
   c) Rename the sphere *Eyeball*.
3. Create an eyelid with a revolution from a curve.
   a) Choose Modeling > Draw Curve and draw a cross-section of the eyelid in Front view.



   b) To make the eyelid surface, choose Modeling > Create Surface > Revolution from Curve.

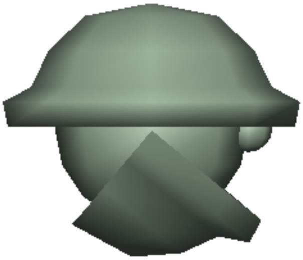


   c) Rename the revolution *Top_eyelid*.
4. Duplicate and scale the eyeball to create a pupil:
   a) Select the eyeball and press **CTRL+D**.
   b) In Front view, translate it towards the right. Its position should be 0.6, 0, 0.
   c) Scale it down to approximately 0.25, 0.25, 0.25.

d) Rename it *Pupil*.

e) Expand the sphere geometry parameters and set Horizontal subdivisions to 6 and Vertical subdivisions to 4. This makes the sphere slightly lighter with no great loss of detail due to its small size.

5. Create the bottom eyelid with Revolution from Curve. For the bottom eyelid, you could draw a new curve. You can also reuse the previous curve:
   a) Select the eyelid curve, and choose Modeling > Create Surface > Revolution from Curve. Click on RevolutionGeometry in the Parameter Editor and set Angle to 180 and Subdivisions to 4.
   b) Select the revolution in the World Explorer and rename it *Bottom_eyelid*.
6. Rotate the bottom eyelid into place.
7. Rotate it using all viewports. An orientation value of (135 -89 0) creates an open lower eyelid.



8. Turn off the curve's visibility.
9. Select all four surfaces (without the curve) and group them. Rename the group *Eye*.

Adding constraints

Use a pose constraint to attach the pupil to the eye, then add a direction constraint between the eyeball and the cursor so that the eye follows the surfer's cursor.

1. Add a pose constraint between the pupil and the eyeball:
   a) Select the pupil and **SHIFT**-select the eyeball.
   b) Turn on the Unlock button.
   c) Choose Relations > Pose Constraints.
   d) Turn off the Unlock button.
2. Create a direction constraint of the eye towards the cursor:
   a) In the World Explorer, multi-select the eyeball and then the WebCursor, in that order.

b) Choose Relations > Direction Constraint.

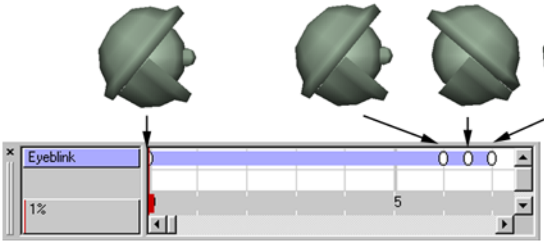
c) Expand the DirectionCns parameters under the eyeball.
The constrained direction is the X axis (1, 0, 0), which is fine since you want it pointing towards the camera.

d) Change the Cursor depth factor to 30%. This places the cursor between the eye and the surfer (represented by his cursor).

e) Switch to Browse mode and move the cursor around. Notice the eyeball, including the pupil, follows the cursor.

Animating the eyelid

Create an animation so that the eye blinks every 7 seconds.



1. Create an animation of the top eyelid blinking:
   a) Turn off playback and go to time 0.
   b) Select the eyelid and rotate it counter-clockwise in Front view so that the eye is fully open.
   c) With the Rotate button turned on and Top_eyelid still selected, turn on the Record button.
   An orientation animation sequence appears in the Sequencer.

2. Save other keyframes:
   a) Go to time 6 and press **CTRL+K** to add a key.
   A key is recorded with the eyelid open at 6 seconds.
   b) Go to time 7 and press **CTRL+K** to add another key.
   A key is recorded with the eyelid open at 7 seconds.
   c) Go to time 6.5 and rotate Top_eyelid shut.
   Since Record is on, a key is automatically recorded when you change the orientation parameter. This key records that the eyelid is shut at 6.5 seconds.
   d) Rename the animation. In the World Explorer, expand Top_eyelid, click AnimationOri twice and type *Eye_blink*.

    e) In the Parameter Editor, set Interpolation to Linear and turn on Cycle.
       The eye blinks every 7 seconds. It is open from 0 to 6 seconds, closed at 6.5, and then open at 7. The animation then cycles back to the beginning.

3. Switch to Browse mode to preview the result. The eye blinks every 7 seconds while following the cursor.

Creating the other eye from a duplicate

To create the other eye, duplicate the geometry, and then create the direction constraint and the eyelid animation.

1. Duplicate the elements of the eye group and create a new group:
    a) In the World Explorer, **ALT**-click the eye group icon.
       Expand the group to see that all four elements are selected.
    b) Choose Modeling > Duplicate or press **CTRL+D**.
       A copy of each element appears in the World Explorer above the group.
    c) With the copies still selected, press **CTRL+G** to group them.
    d) Rename the group *Eye_2*.
2. Apply the same constraints and animation to Eye_2 as Eye_1. Link the pupil to the eyeball with a pose constraint, and link the eyeball to the cursor with a direction constraint. Then create a blink animation for the eyelid. For more information, see “Animating the eyelid” on page 102.

Positioning and constraining the eye groups

Place the eyes on the cactus and attach them to the upper backbone so that they follow the movement of the head.

1. Turn on visibility for both the cactus group and the limbs group.
2. Select both eye groups and translate, scale, and rotate them into position.
3. Constrain the eye groups to the backbone:
   a) Turn on the Unlock button before you apply the constraint.
   a) In the World Explorer, multi-select one eye group and then the upper backbone, in that order.
   b) Choose Relations > Pose Constraint.
   c) Repeat the same procedure for the second eye.
   d) Turn off the Unlock button, and translate different parts of the cactus to test the character.

Adding materials

AXEL assigns a default material to objects, which explains the rather unhealthy look of our cactus. You can apply a different material to each object as soon as you build it or you can link one material to several objects. Linking materials is practical if you have many objects that you may want to change all at once, for example, plane seats. In this lesson, you created the objects first; now link them to materials and their textures.

1. In the World Explorer, expand Resources / Materials, and select Default Material.
   In the Parameter Editor, notice the default material is gold. To change the color, click the color chip and select a color in the Color Editor.
2. Add a material for the eyeballs:
   a) In the World Explorer, expand both eye groups, multi-select both eyeballs, and choose Visualization > Add Material.
   b) Pick the pure white color and click OK.
   Both eyeballs are now white and two new materials appear in the Resource bank. It is practical to have two copies of the same material if you want to change them individually.
   c) Rename the materials *eyeball_material_1* and *eyeball_material_2*.
3. To change the material of one eyeball, select it, choose Visualization > Edit Material Color, select a new color in the Color Editor, and click OK.

   Note: To change an object's color, you can also expand its parameters, select its material parameter, then click the color chip in the Parameter Editor.
4. Add a material to one of the pupils:
   a) Select a pupil and choose Visualization > Add Material, then select a dark green color.
   b) Rename the new material *green_eyes*. Notice you can select the material in two places in the World Explorer, under the object's material or in the Resource bank.

5. Link the other pupil to the *green_eyes* material. Multi-select the other pupil and the green_eyes material, and choose Visualization > Link Material or press **CTRL+L**.

   Now both pupils are linked to only one material. If you modify the material, both pupils will change.

6. Link a material to the thorns:
   a) Select a thorn and add a light green material to it. Rename the material *thorn_green*.
   b) To link all other thorns to the thorn_green material, select all the other thorns and the thorn_green material and press **CTRL+L**.

Note: If you had applied a material to the original thorn and then created instances, all new thorns would be linked to the same material. If you had applied a material to the original thorn and then created duplicates, all new thorns would have a different material with the same color values.

Adding textures

A texture is a rasterized image that is applied over a material on an object.By default, textured are projected through a plane, like a movie screen in front of the object. For the cactus, replace the default projection with a cylindrical projection so that the texture image wraps around the cactus body.

1. Add a texture to the cactus body:
   a) Select the cactus_body and choose Visualization > Add Texture.
   b) Select *cactus_skin_64.bmp* from the Documentation / Tutorials / Textures folder. AXEL automatically generates a material that contains the texture. If you expand the cactus body in the World Explorer, you will notice a new material. Expand it to see the Texture Map (cactus_skin_64). Rename that material *cactus_material*.

      The texture is projected onto the surface through a flat screen in front of the cactus.

About Texture Images

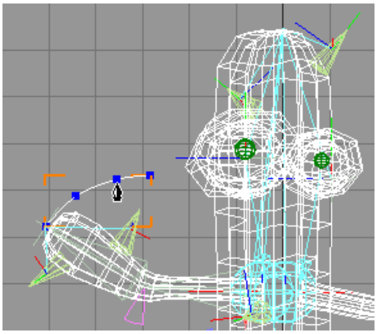
Open GL supports powers of 2 in terms of pixels, so textures are usually saved in formats of 16, 32, 64 or 128 pixels. The size of texture files significantly affects download time and performance. If the texture file is too big, it slows down the download and the refresh rate, but if it is too small compared to the size of the object that it is covering, it is stretched so much that you can see the individual pixels in the texture image. To compromise between performance and quality, a 64 x 64 pixel image was used for the cactus.

2. Apply the texture through a cylindrical texture map so that it wraps around the cactus:
   a) Select cactus_body and choose Visualization > Texture Projection > Cylindrical.
      The texture changes appearance, and a texture projector appears in the viewports and World Explorer.
   b) Scale and translate the texture projector so that it fits tightly around the cactus body. If you scale down the texture projector vertically, the texture is tiled along the length of the cylinder.
   c) You can keep the texture projector in the World as long as you want to modify the texture's appearance. But if you want the texture to be attached to the object so that it bends and stretches with the object, then delete the texture projector. In this lesson, since the texture must deform with the cactus_body, place the texture in a position you like, and then delete the texture projector.
3. Link the arms to the cactus material:
   a) Multi-select both arms and the cactus material.
   b) Choose Visualization > Link Material or press **CTRL+L**.
      The texture appears on the arms. Since they are now both linked to the cactus material that contains the cactus texture, they also receive that texture.
4. Link the eyelids to the cactus material.
5. Translate the end bones of the arms or the backbone to move the cactus. Notice the texture follows and deforms with the geometry.

Waving goodbye using IK and path animation

Use inverse kinematics (IK) to make the arm wave by constraining the end bone on a path and animating it.

1. Create a path animation:
   a) In the Right viewport, draw a curve next to the end of the right arm.



   b) Set the curve steps to 10, and rename the curve *Hello_curve*.

   c) Multi-select the end bone of right_arm and then the curve, in that order.

   d) Choose Animation > Add Path Animation.
   The bone follows the curve and an animation over 10 seconds is created by default. If you play the animation from time 0, you will see the arm go from the beginning to the end of the curve over 10 seconds.

2. Add a key to modify the default path animation:

   a) Select AnimationParam in the Sequencer. At time 5, press **CTRL+K** to add a key.

   b) In the Parameter Editor, double-click the keys. Correct the values to the following:

0.00	0.00
5.00	100.00
10.00	0.00

   At seconds 0 and 10, the arm will be at the beginning of the curve (0%) and at second 5 at the end of curve (100%).

   c) Select AnimationParam again in the Sequencer and press **P** to play the animation.

   d) Turn on the cycle function in the Parameter Editor.
   The arm is crawling along.

3. Change the timing:

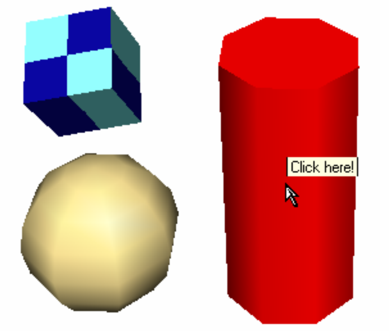
   a) You can either click and drag the keys in the Sequencer to 0, 1, and 2 seconds respectively, or drag the Time scale in the Parameter Editor to 5 to compress the animation length.
   Depending on the location of the arm, you may want to adjust Hello_curve by translating it or moving points to ensure a pleasant wave!

4. Choose Interaction > Handles > Orbit Camera to be able to orbit with the left mouse button, and switch to Browse mode to enjoy your work.

9 Interaction

Switching Reactions

In this lesson, you use reactions to swap the shapes and materials of two objects. You also add a tooltip to inform surfers of what will happen when they click the page.



Need help? — Refer to the *Switching_complete.axel* file in the AXEL / Documentation / Projects folder.

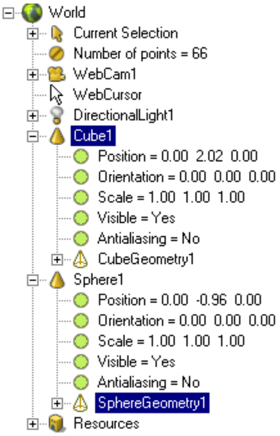
In this lesson:

Switching shapes

Set up an interaction so a surfer can click a red cylinder to change a cube into a sphere.

1. Create a cube and a sphere, and then move them apart:
   a) Choose Modeling > Preset Surfaces > Cube.
   b) Choose Modeling > Preset Surfaces > Sphere.
   c) Translate the sphere above the cube so that both are visible.

2. Add a reaction to the cube so that it switches its geometry with the sphere:
   a) In the World Explorer, expand the cube and the sphere.
   b) Multi-select the cube and the sphere’s geometry, in that order.



```
World
   Current Selection
   Number of points = 66
   WebCam1
   WebCursor
   DirectionalLight1
   Cube1
      Position = 0.00  2.02  0.00
      Orientation = 0.00  0.00  0.00
      Scale = 1.00  1.00  1.00
      Visible = Yes
      Antialiasing = No
      CubeGeometry1
   Sphere1
      Position = 0.00  -0.96  0.00
      Orientation = 0.00  0.00  0.00
      Scale = 1.00  1.00  1.00
      Visible = Yes
      Antialiasing = No
      SphereGeometry1
   Resources
```

   c) Choose Interaction > Reactions > Set Geometry.
   A SetGeometry reaction appears in the Interaction Editor.

Tip: To make more room for the Interaction Editor, close the Sequencer. The Interaction Editor expands to take the Sequencer’s space.

3. Create a red cylinder to use as a sensor:
   a) Choose Modeling > Preset Surfaces > Cylinder.
   b) Translate the cylinder under the cube.

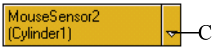
c) In the Parameter Editor, click Cylinder Geometry, then turn on Top capping, End capping, and Crease control.

d) Make the cylinder red. Choose Visualization > Edit Material Color and select a red color chip.

e) With the cylinder still selected, choose Interaction > Sensors > Mouse.
The MouseSensor (Cylinder) appears in the Interaction Editor.

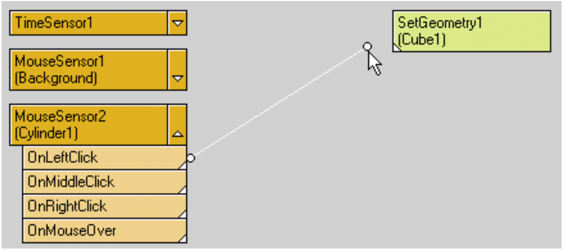
4. Link the sensor to the reaction:

a) In the Interaction Editor, click the arrow in the MouseSensor (Cylinder) to expand it.



Click to expand sensor

b) Click and drag from On Left Click to SetGeometry to link the sensor to the reaction.



5. Switch to Browse mode to test the interaction. When you click the cylinder, the cube changes to a sphere.

6. Switch back to Author mode to continue working in the World.

7. The sphere must stay in the World as a reference but it can be hidden. Select the sphere and turn off its visibility in the Parameter Editor.

8. Edit the trigger between the sensor and the reaction:

a) In the Interaction Editor, multi-select OnLeftClick and SetGeometry, or select the line that links them.
The trigger parameters appear in the Parameter Editor.

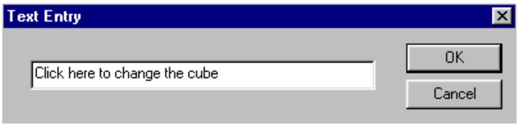
b) Set Trigger mode to Toggle Start/Stop.

c) Switch to Browse mode to test the interaction. One click changes the cube into a sphere; a second click changes it back into a cube.

Adding a tooltip

Add a tooltip reaction to tell surfers about the interaction.

1. Switch to Author mode.
2. Select the cylinder and choose Interaction > Reactions > Tooltip Text.
3. Type the message shown in the following illustration, and then click OK.



Click here to change the cube

4. In the Interaction Editor, expand the MouseSensor (Cylinder).

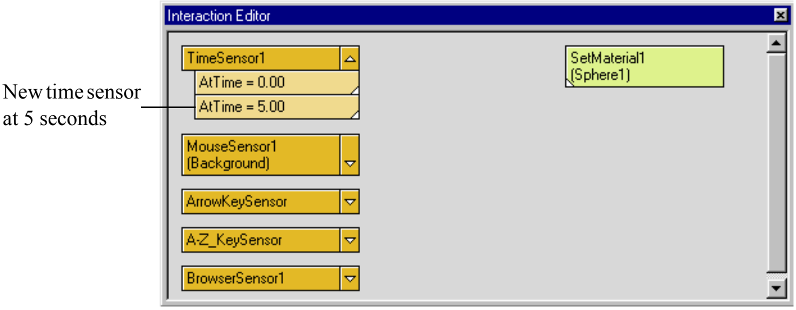
   Notice that OnMouse Over is automatically linked to the Tooltip Text reaction.
5. Switch to Browse mode to test the interaction.
6. To change the text, switch to Author mode, select the Tooltip Text reaction in the Interaction Editor, and adjust its parameters in the Parameter Editor.

Switching material color

Set up an interaction so that a cube's color changes 5 seconds after a surfer opens the web page.

1. Choose Modeling > Preset Surfaces > Cube.
2. Translate the cube to the right of the other objects.
3. To duplicate the cube, select it and choose Edit > Duplicate or press **CTRL+D**. The new cube is slightly offset. Translate the cube so it covers the first cube.
4. Rename the bottom cube *Blue_cube* and the top one *Red_cube*.
5. To make Blue_cube blue, select it and choose Visualization > Edit Material Color. In the Color Editor, select a blue color chip and click OK.
6. To make Red_cube red, choose Visualization > Edit Material Color. In the Color Editor, select a red color chip and click OK.

7. Add a reaction to Blue_cube to switch its material with another:
   a) In the World Explorer, select Blue_cube, expand Red_cube, and then multi-select its material.
   b) Choose Interaction > Reactions > Set Material.
   In the Interaction Editor another reaction is created. When this reaction is triggered, Blue_cube will switch from blue to red.
8. Create a time sensor to make the cube change color 5 seconds after the web page is opened:
   a) Re-display the Sequencer if hidden (press **ALT+4**), and turn off playback.
   b) Go to time 5 by typing *5* in the Time field on the main toolbar or moving the playhead to 5 in the Sequencer.
   c) Choose Interaction > Sensors > Time.
   To see the new sensor, expand the TimeSensor in the Interaction Editor.



   d) To link the sensor to the reaction, click and drag the cursor from AtTime=5 to Set Material, or multi-select the two and press **CTRL+L**.
   e) Switch to Browse mode to test the interaction.
   After 5 seconds, the blue cube turns red. Switch back to Author mode to continue working in the World.

Switching textures

Set up an interaction so that the texture on the cube changes 5 seconds after a surfer opens the web page. Use the Set Material reaction to switch from one texture to another.

1. Add a checkerboard texture to the blue cube:
   a) Select the blue cube and choose Visualization > Add Texture.
   b) In the Open dialog, navigate to Documentation / Tutorials / Textures, select *CheckerBoard.bmp* and click Open.
   The checkerboard appears on the blue cube. The default texture mode is Multiply. In Multiply mode, the texture is blended with the material color.

c) Make the texture replace the material color. In the World Explorer, expand Blue_cube / Material / Texture Settings, then select Replace from the Texturing mode list in the Parameter Editor.

2. Add a wood texture to the red cube:

   a) Select the red cube and choose Visualization > Add Texture.

   b) In the Open dialog, navigate to Documentation / Tutorials / Textures, select *Wood_cylinder.bmp* and click Open.
   The wood texture appears on the red cube. You may also want to change the texture mode to Replace.

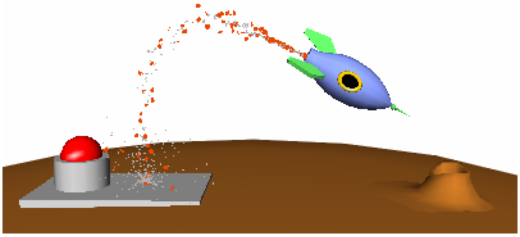
3. Turn off the visibility of the red cube and switch to Browse mode to test the interaction.

10 Interaction

Rocket Launch

You can liven up a web page by making the surfer trigger dynamic events. Events are linked to sensors like a mouse click or elapsed time. When triggered, these sensors set off one or more reactions, like sounds, color changes, and animated movement.

In this lesson, you apply a sensor to a red button that, when clicked, triggers a reaction that launches a rocket.



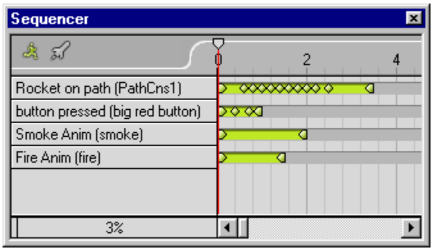
Need help? — Refer to the *Rocket_completed.axel* file in the AXEL / Documentation / Tutorials folder.

In this lesson:

Creating sensors and reactions to launch a rocket

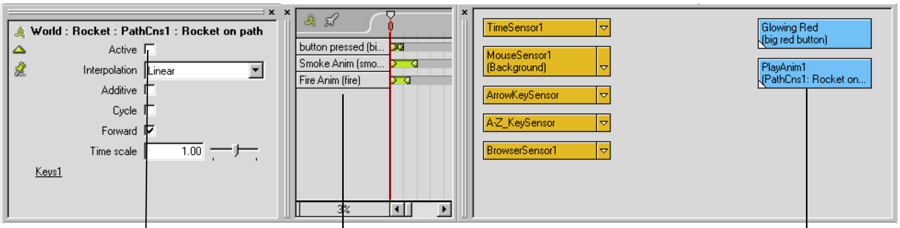
In this part of the tutorial, turn rocket animations and sound files into reactions. Then link them to a button sensor so they can be triggered by the surfer.

1. In the Documentation / Tutorials folder, open the *Rocket_start* file:
   An AXEL file displays a world containing a rocket, a mountain, and a launch button.
2. Switch to Browse mode to view the animation, then go back to Author mode.
3. The Sequencer contains four animated sequences, each of which defines a specific type of rocket and button animation. These animations automatically play as soon as the surfer opens the web page. What you want to do is turn these animations into reactions so they play only when you want them to:



Rocket on path (PathCns1)
button pressed (big red button)
Smoke Anim (smoke)
Fire Anim (fire)

   a) In the Sequencer, select the Rocket On Path sequence.
   b) Choose Interaction > Reactions > Play Animation.
   The rocket animation is removed from the Sequencer and appears as a reaction in the Interaction Editor. The animation is now "on hold", waiting to be triggered by the sensor you define in the next step.



World : Rocket : PathCns1 : Rocket on path
Active □
Interpolation Linear
Additive □
Cycle □
Forward ☑
Time scale 1.00
Keys1

button pressed (bi...
Smoke Anim (smo...
Fire Anim (fire)
3%

TimeSensor1
MouseSensor1 (Background)
ArrowKeySensor
A-Z_KeySensor
BrowserSensor1
Glowing Red (big red button)
PlayAnim1 (PathCns1: Rocket on...

Animation no longer active

Animation disappears from Sequencer

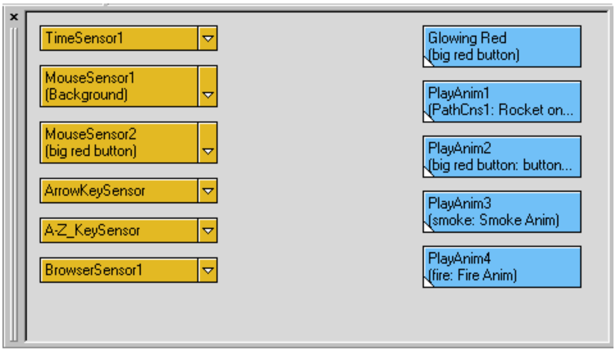
Animation appears as reaction in Interaction Editor

   c) Shift-select the remaining three animations in the Sequencer, and choose Interaction > Reactions > Play Animation.
   Three more play animation reactions are created.

4. Turn the red launch button into a sensor and link it to the reactions you just created. This way, the button will trigger the reactions each time the surfer clicks on it:

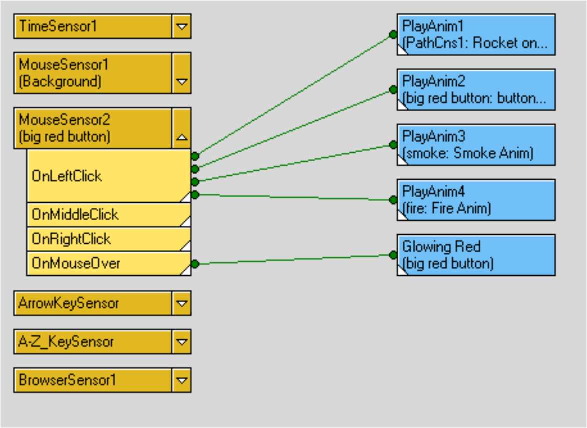
   a) Select the red button.

   b) Choose Interaction > Sensors > Mouse.
   A mouse sensor appears as MouseSensor2 in the left column of the Interaction Editor. Note how a number of sensors have already been created in the left column. They can come in handy if you wish to make a sensor out of keyboard keys, time, a Browser download, or a mouse click in the background.


TimeSensor1
MouseSensor1 (Background)
MouseSensor2 (big red button)
ArrowKeySensor
A-Z_KeySensor
BrowserSensor1
Glowing Red (big red button)
PlayAnim1 (PathCns1: Rocket on...
PlayAnim2 (big red button: button...
PlayAnim3 (smoke: Smoke Anim)
PlayAnim4 (fire: Fire Anim)

   c) Expand MouseSensor2 (Big Red Button), click on the lower right corner of OnMouseOver, and drag a link to the Glowing Red reaction in the column to the right (Glowing Red is a Set Material reaction that gives the red color a glowing effect).

   d) Click on OnLeftClick and drag a link to the four remaining reactions in the right column.

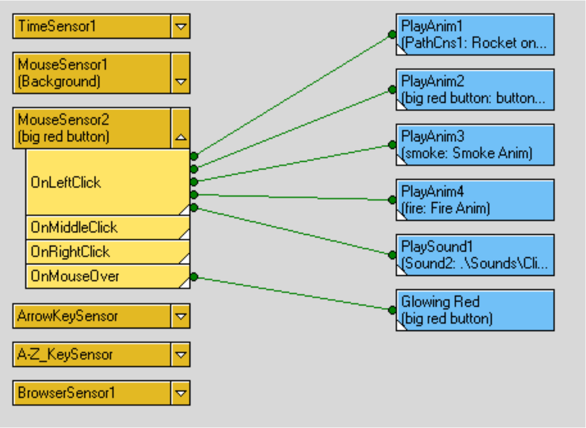

TimeSensor1
MouseSensor1 (Background)
MouseSensor2 (big red button)
OnLeftClick
OnMiddleClick
OnRightClick
OnMouseOver
ArrowKeySensor
A-Z_KeySensor
BrowserSensor1
PlayAnim1 (PathCns1: Rocket on...
PlayAnim2 (big red button: button...
PlayAnim3 (smoke: Smoke Anim)
PlayAnim4 (fire: Fire Anim)
Glowing Red (big red button)

5. Trigger a sound to play each time the surfer left-clicks the button:

   a) Make sure everything is deselected, then choose Interactions > Reactions > Play Sound.

   b) In the Browser, navigate to the Sounds folder and open *Click.wav.*
   A new PlaySound reaction appears in the right column of the Interaction Editor.

   c) At MouseSensor2, click on OnLeftClick and drag a link to the PlaySound reaction.


TimeSensor1
MouseSensor1 (Background)
MouseSensor2 (big red button)
OnLeftClick
OnMiddleClick
OnRightClick
OnMouseOver
ArrowKeySensor
A-Z_KeySensor
BrowserSensor1
PlayAnim1 (PathCns1: Rocket on...
PlayAnim2 (big red button: button...
PlayAnim3 (smoke: Smoke Anim)
PlayAnim4 (fire: Fire Anim)
PlaySound1 (Sound2: .\Sounds\Cli...
Glowing Red (big red button)

6. Switch to Browse mode and left-click the red launch button.

   By turning the red button into a sensor and linking it to a series of reactions (playback of animation and sound files), you have created a simple interaction that can be triggered by the surfer at any time. As the mouse passes over the button, it sets off the glowing red reaction. The left mouse click sets off the remaining sound and animation reactions.

7. As a finishing touch, create a handle that lets the surfer orbit the World. Choose Interaction > Handles >OrbitWebCam.

   MouseSensor1 (background) is automatically linked to a newly created reaction called OrbitWebCam1. Now, if the surfer left-clicks and drags on any part of the World that is not already a sensor, the WebCam orbits.

Editing Reactions, Sensors, and Triggers

Now that you have built your interaction, you will notice a few things you may want to change:

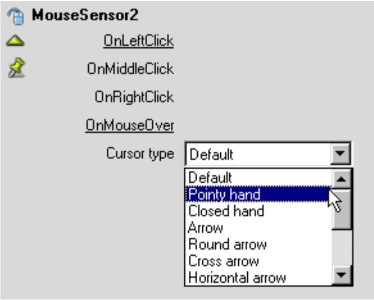
- The default cursor appears when it passes over the button sensor.

- Once the cursor passes over the button, it remains stuck in its active color.

- The surfer can orbit horizontally and vertically, but you want him/her to only be able to orbit horizontally.

Fix these little glitches by editing your World's reactions, sensors, and triggers.

1. First, edit the cursor type so it changes to a more appropriate shape as it passes over the button sensor:
   a) In the Interaction Editor click on Mouse Sensor2 (Big Red Button).
   The sensor's parameters are displayed in the Parameter Editor.



MouseSensor2

OnLeftClick

OnMiddleClick

OnRightClick

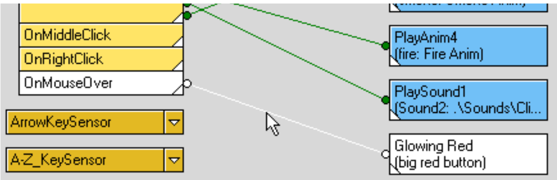
OnMouseOver

Cursor type: Default

   b) From the Cursor type list, select Pointing hand.
   The cursor now turns into a pointing hand as it passes over the sensor to indicate that the button is interactive.

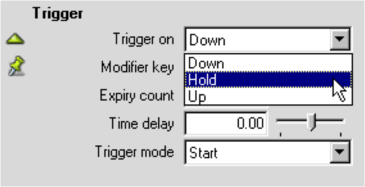
2. Next, edit a trigger so the button material turns an active color only when the cursor is directly over it:

   a) In the Interaction Editor click on the link (the trigger) between Mouse Sensor2 (Big Red Button) and Glowing Red (Big Red Button).



The trigger's parameters are displayed in the Parameter Editor.

b) From the Trigger on list, select Hold.



Trigger

Trigger on: Down

Modifier key

Expiry count

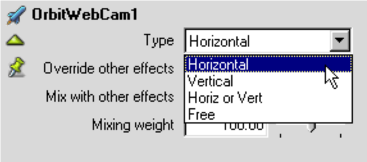
Time delay: 0.00

Trigger mode: Start

3. Now modify the handle so the surfer can only orbit along a horizontal plane:

a) In the Interaction Editor, click the OrbitWebCam sensor.

b) In the Type list of the Parameter Editor, select Horizontal.



OrbitWebCam1

Type: Horizontal

Override other effects

Mix with other effects

Mixing weight: 100.00

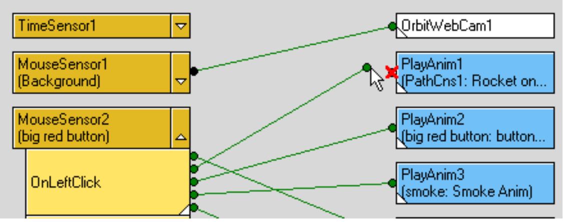
c) Switch to Browse mode and try to orbit around the World. You will only be able to orbit along the X axis.

Grouping reactions

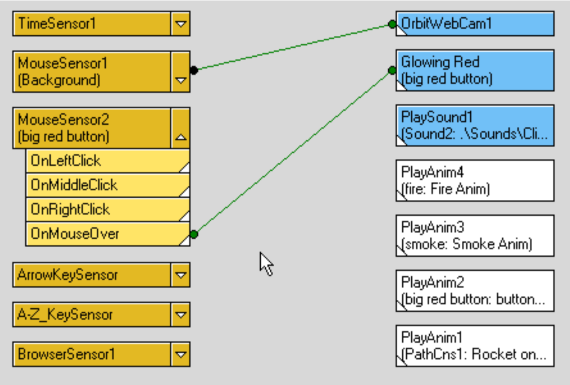
You can group reactions to edit them as a single unit and allow more control over the sequence of interactions.

1. Group the play animation reactions:

   a) In the Interaction Editor, break the links of each animation and sound reaction by dragging the link anchor points away from the reactions or by selecting the trigger and pressing **CTRL+U**. (Be sure not to unlink the trigger between the MouseOver and the Glowing Red reaction.)


TimeSensor1
MouseSensor1 (Background)
MouseSensor2 (big red button)
OnLeftClick
OrbitWebCam1
PlayAnim1 (PathCns1: Rocket on...
PlayAnim2 (big red button: button...
PlayAnim3 (smoke: Smoke Anim)

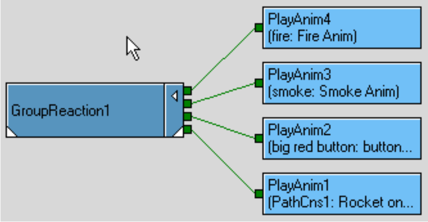
   b) Multi-select the four animation reactions.


TimeSensor1
MouseSensor1 (Background)
MouseSensor2 (big red button)
OnLeftClick
OnMiddleClick
OnRightClick
OnMouseOver
ArrowKeySensor
A-Z_KeySensor
BrowserSensor1
OrbitWebCam1
Glowing Red (big red button)
PlaySound1 (Sound2: .\Sounds\Cli...
PlayAnim4 (fire: Fire Anim)
PlayAnim3 (smoke: Smoke Anim)
PlayAnim2 (big red button: button...
PlayAnim1 (PathCns1: Rocket on...

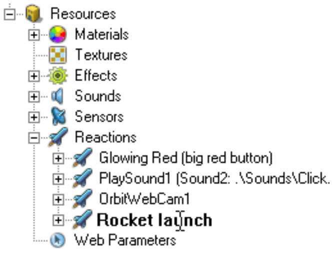
c) Press **Ctrl+G** to group the selected reactions.

Note: Another way of accomplishing the previous steps would be to select all four linked reactions and group them. Creating a group of reactions would automatically unlink them.

The reactions are grouped into a blue box called GroupReaction1. Expand it by clicking on the white triangle at the top of the box.



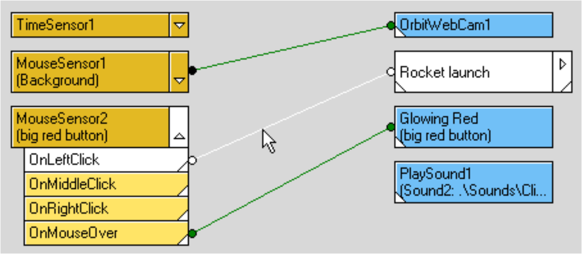
d) In the World Explorer, expand Resources / Reactions, press **F2**, and rename GroupReaction1 *Rocket Launch*.



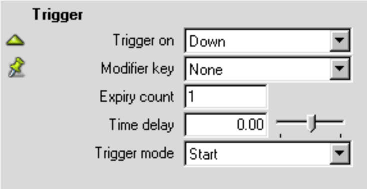
2. Now that you have created the rocket launch group, edit its reactions so that its animation plays only once:

a) In the Interaction Editor, link OnLeftClick of MouseSensor2 (Big Red Button) to the Rocket Launch reaction group.

b) Click the trigger link you just created.



c) In the Expiry count list of the Parameter Editor, change the value to 1.



Trigger

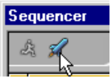
Trigger on	Down
Modifier key	None
Expiry count	1
Time delay	0.00
Trigger mode	Start

Now, the rocket animation will only play once.

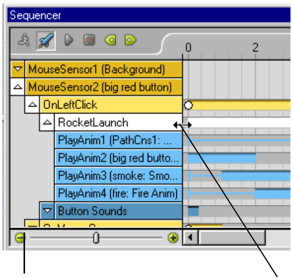
Offsetting reactions

Adjust the timing of the animation sequences in the Rocket Launch group so they take place at different times.

1. In the Sequencer, click on the rocket icon to switch to Interaction view.



2. Enlarge the Sequencer window.
3. In the left column, expand MouseSensor2 (Big Red Button) until you can see all the reactions in Rocket Launch.

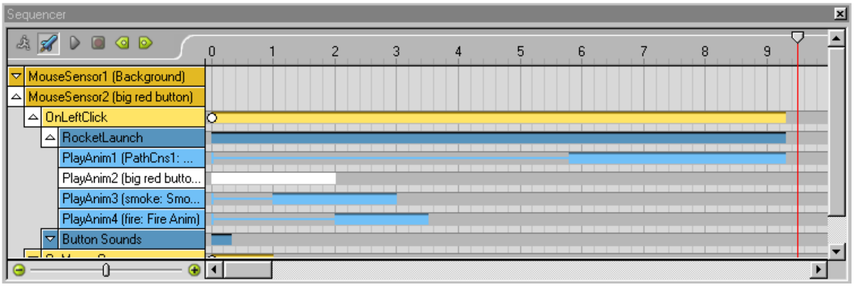

Sequencer
0
2
MouseSensor1 (Background)
MouseSensor2 (big red button)
OnLeftClick
RocketLaunch
PlayAnim1 (PathCns1: ...
PlayAnim2 (big red butto...
PlayAnim3 (smoke: Smo...
PlayAnim4 (fire: Fire Anim)
Button Sounds

Zoom bar

Drag this line to display full sequence names

4. Drag the zoom bar to zoom in on the interaction timeline until you can see all elements clearly, then adjust the timing as indicated below:
   a) Click and drag PlayAnim3(smoke:Smoke Anim) to the 1-second mark.
   b) Click and drag PlayAnim4 (fire:Fire Anim) to the 2-second mark.

c) Click and drag PlayAnim1 (Path cns:Rocket On Path) to the 4.5-second mark.


Sequencer
0
1
2
3
4
5
6
7
8
9
MouseSensor1 (Background)
MouseSensor2 (big red button)
OnLeftClick
RocketLaunch
PlayAnim1 (PathCns1: ...
PlayAnim2 (big red butto...
PlayAnim3 (smoke: Smo...
PlayAnim4 (fire: Fire Anim)
Button Sounds

5. Switch to Browse mode and click the big red button to trigger the animation.

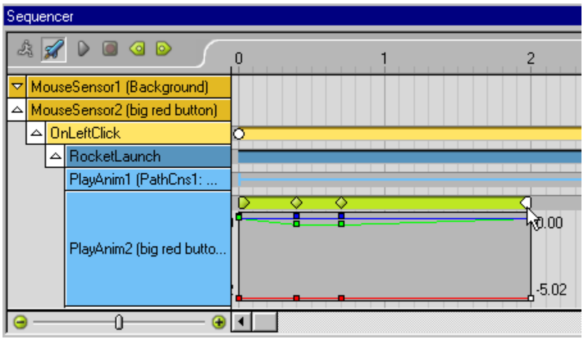
   Notice how the reactions in the rocket launch have changed. By changing the timing of the animations within a group, you can create elaborate scenarios. You could, for example, add a reaction to the group that causes another sensor to appear, creating more interaction possibilities.

Editing the timing of reactions

Even after you convert animations into reactions, you can still edit their keys and values. Now edit the animation of the red button so that it takes longer to pop back to its original position after being clicked.

1. In the Sequencer, double-click PlayAnim2(Big Red Button:button pressed).
   The button's sequence expands to display its Curve Editor.

2. Click and drag the button's last key to the 2-second mark.


Sequencer
0
1
2
MouseSensor1 (Background)
MouseSensor2 (big red button)
OnLeftClick
RocketLaunch
PlayAnim1 (PathCns1: ...
PlayAnim2 (big red butto...
0.00
-5.02

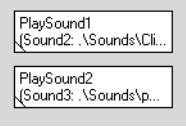
3. Switch to Browse mode to see the results.

Changing the playing order of sounds

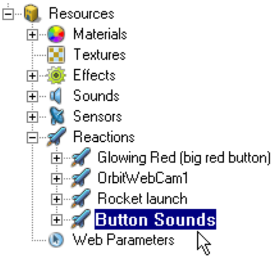
To add variety to your World or provide additional information, you may want to associate more than one sound to a sensor. If you do, you can cause different sounds to play, either together, in a specific order, or at random when the sensor is triggered.

Here, attach two sound reactions to the big red button sensor, and have them play in a specific order. The first sound will play when the surfer clicks the button for the first time, to indicate the rocket's liftoff. A second, different sound will play if the surfer clicks the button a second time to indicate that the liftoff has already taken place.

1. Choose Interaction > Reactions > Play Sound, and in the Browser under the Sounds folder, open *Pop.wav*.
2. In the Interaction Editor, multi-select Playsound1 and Playsound2 reactions. Be sure to select the reactions in that order because that is the order in which you want them to play.

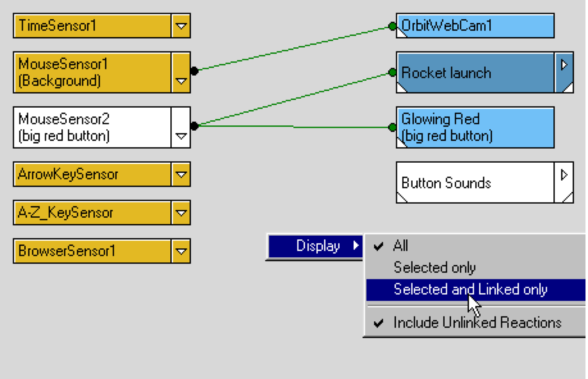


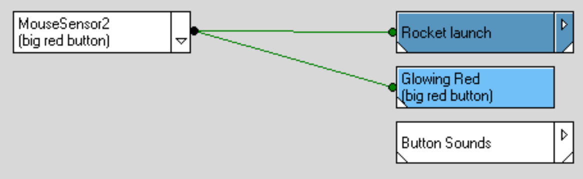
3. Press **CTRL+G** to group the sounds.
4. In the World Explorer, rename GroupReaction1 *Button Sounds*.



5. In the Interaction Editor, multi-select MouseSensor2 and Button Sounds.

6. To focus on just your selected elements, right-click anywhere in the editor and choose Display > Selected and Linked Only.



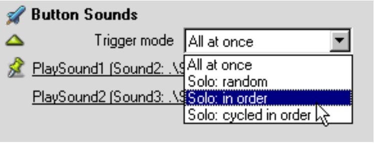


Later, you can revert to viewing all the elements by right-clicking in the editor and choosing Display > All.

7. Expand MouseSensor2 and link OnLeftClick to Button Sounds.

8. Select Button Sounds.

9. In the Parameter Editor, in the Trigger mode list, select Solo: in order.



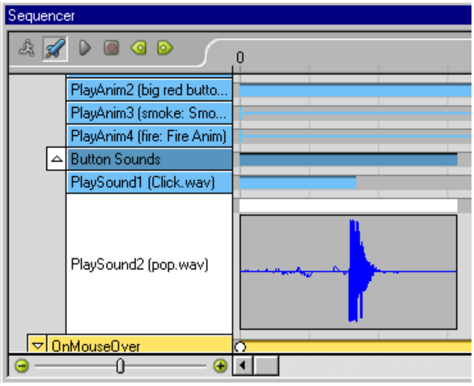
Button Sounds

Trigger mode: All at once

PlaySound1 (Sound2: .\S

PlaySound2 (Sound3: .\S

This causes the two reaction sounds to play in the order in which they were selected. The first sound plays the first time the red button is clicked; the second sound plays on the second click. All subsequent clicks on the button will produce no sound at all.



Sequencer

PlayAnim2 (big red butto...

PlayAnim3 (smoke: Smo...

PlayAnim4 (fire: Fire Anim)

Button Sounds

PlaySound1 (Click.wav)

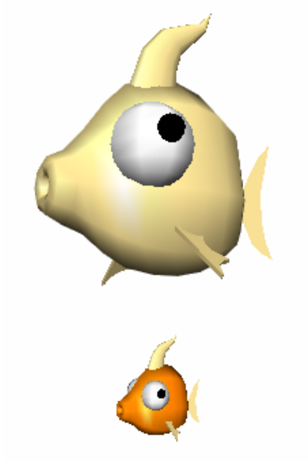
PlaySound2 (pop.wav)

OnMouseOver

By double-clicking on a *.wav* file in the Sequencer, you can view a graphic representation of the sound, which can be useful when you want to synchronize other reactions with it.

11 Revolution, Extrusion, Manipulating Points
Fish — Modeling

In this part of the tutorial, you model the fish by creating the various body parts using AXEL's modeling tools. Finally you'll group these elements, then duplicate them to create another fish.



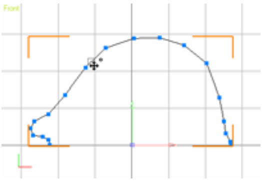
Need help? — Refer to the *Fish_modeling_complete.axel* file in the AXEL / Documentation / Tutorials folder.

In this lesson:

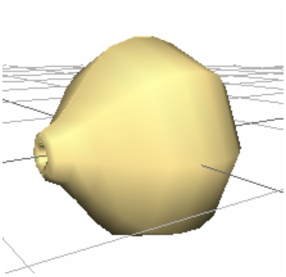
Modeling the main body

Create the body using a revolution from curve, then refine the shape by editing points on the revolution surface.

1. Open the *fish_modeling_start.axel* file in the Documentation/Tutorials folder.
   An AXEL file displays a world containing a curve.



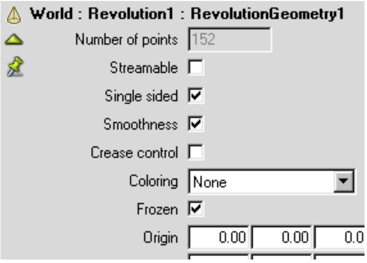
2. Select the curve and choose Modeling > Create Surface > Revolution from Curve.



   A spherical fish body is created from the curve.

3. Rename the revolution you just created *Fish body*.

4. Edit points on the fish body to make it look more like a fish:
   a) Select Fish body and in the Parameter Editor click RevolutionGeometry.
   b) When you start manipulating points, the Frozen parameter turns on.



World : Revolution1 : RevolutionGeometry1

Number of points 152

Streamable □

Single sided ☑

Smoothness ☑

Crease control □

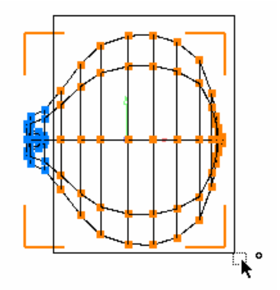
Coloring None

Frozen ☑

Origin 0.00 0.00 0.0

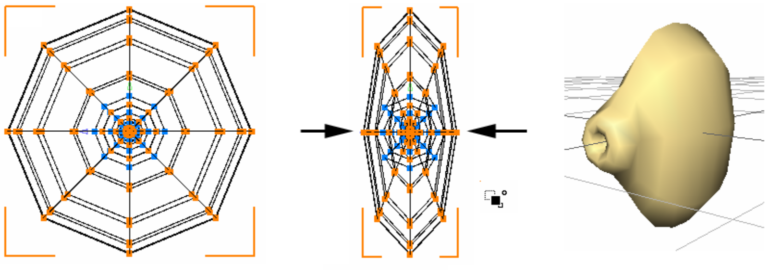
   This indicates that the link between the curve and the revolution is broken. Now if you edit the curve, the fish is not affected.

c) Switch to Point mode and rectangle-select all the points except those to the far left.

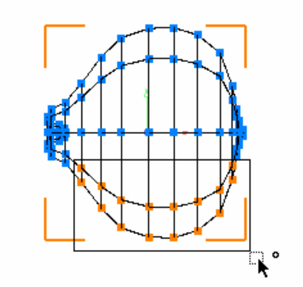


)

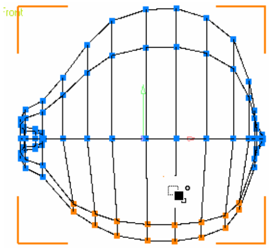
d) Hold the **E** key and in the Right viewport, drag the points with your middle mouse button to flatten the fish (except its mouth).



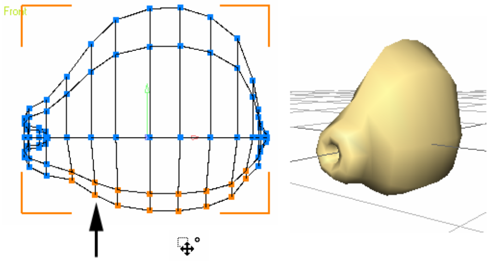
e) In the Front viewport rectangle-select the lower points of the fish as shown in the illustration below.



f) Holding the **E** key, scale the points until they form a less rounded shape.



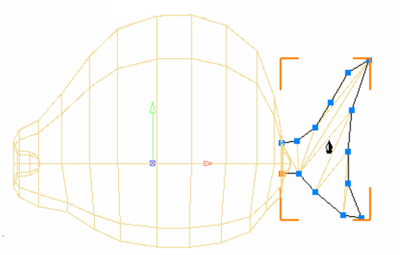
g) Now hold the **T** key and translate the points upward towards the center of the fish.



Using a polygon for a tail

Draw a polygon to use as a tail.

1. Click in the Front viewport and press **ALT+1** to maximize it.
2. Choose Modeling > Draw Polygon and draw a curve similar to the one in the figure shown below.



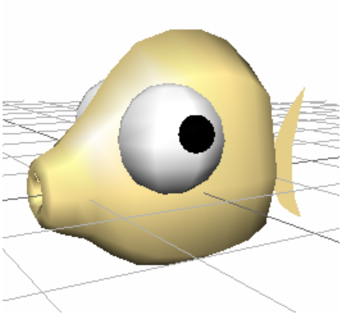
3. Press **ESC** twice to end the polygon and exit from draw polygon mode.

4. Press **3** to switch back to object mode.

Creating eyes from spheres

Create an eye from a group of two spheres, then duplicate the group to create a second eye.

1. Model a fish eye:
   a) Get a sphere by choosing Modeling > Preset Surfaces > Sphere.
   b) Translate the sphere to the desired position for an eyeball. Scale it to an appropriate size.
   c) Add a white material.
   d) Create the eye's pupil by duplicating the sphere (**CTRL+D**), changing its material to blue, then scaling it down to an appropriate size.
   e) Group the eye and the pupil and call it *Eyeball1*.
2. Duplicate the eye and move it to the other side of the fish:
   a) In the World Explorer, **ALT**-click the eyeball group icon to select all the objects in the group, then press **CTRL+D** to duplicate them.
   The duplicated spheres appear as selected in the World Explorer.
   b) Press **CTRL+G** to group the selected spheres and call the new group *Eyeball2*.
   c) Rotate Eyeball2 on its Y axis.
   Notice that the spheres rotate around the center of the group which is at the World origin.
   d) You cannot change the center of the group, but you can move the objects onto the center. Multi-select the spheres, hold **T** and translate them so that the eyeball is at the origin of the World.
   e) Rotate the group and translate it to the other side of the fish.



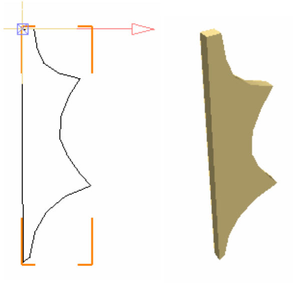
Using Extrusions to create fins

Use a simple extrusion for the lateral fins, then use an extrusion along curve to create a dorsal fin.

1. Create two lateral fins:
   a) In the World Explorer, select Fin curve and toggle on its visibility.

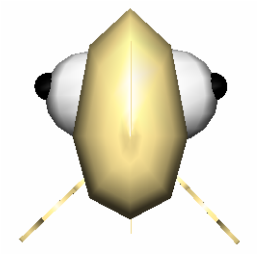
   b) Select Fish body and toggle off its visibility so you can see your fin better.

   c) Select the Fin curve again and choose Modeling > Create Surface > Extrusion from Curve. You now have a fin, but it's too thick to be of any use to the fish. You need to make it thinner.



   d) In the Parameter Editor, click ExtrusionGeometry and set the length to 0.1.

   e) Toggle on visibility of the fish body again, then translate and rotate the fin to its lower right side.



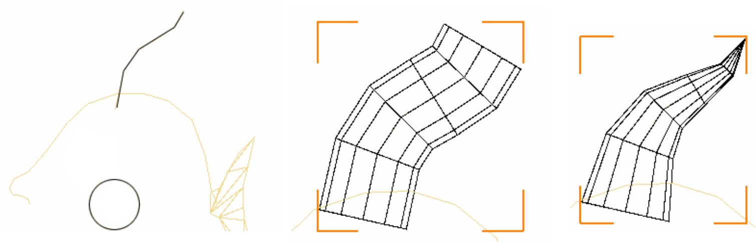
   f) Duplicate the fin, then rotate and translate it to the opposite side of the fish body.

2. You will now create a dorsal fin using AXEL's Extrusion along Curve feature. (Extruding along a curve lets you create a tube-like body, like a steam pipe or a coat hanger.)
   a) Select Fish body and toggle off its visibility so you can see your fin better.

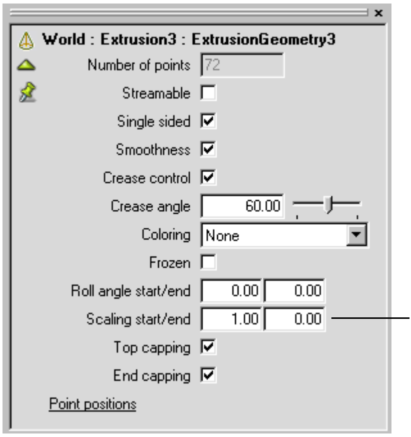
   b) In the World Explorer, make Dorsal fin circle and Dorsal fin path visible.
   The first object defines the shape of the extrusion's cross-section and the second object defines its path.

c) Multi-select Dorsal fin circle and Dorsal fin path, in that order.

d) Choose Modeling > Create Surface > Extrusion along Curve.



e) Click on ExtrusionGeometry and set Scaling start/end to 1.00 and 0.00. The base of the fin starts its widest point, then tapers off into nothing.



World : Extrusion3 : ExtrusionGeometry3

Number of points 72

Streamable □

Single sided ☑

Smoothness ☑

Crease control ☑

Crease angle 60.00

Coloring None

Frozen □

Roll angle start/end 0.00 0.00

Scaling start/end 1.00 0.00

Top capping ☑

End capping ☑

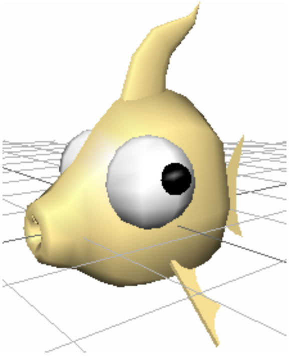
Point positions

Set end scaling to 0

f) In the Parameter Editor, click the up arrow, then set Scale to 1.00, 1.00, 0.30. By scaling on the Z axis, you flatten the fin’s shape.

g) Make the Fish body visible.

3. Group the fish body, fins, tail and eyes:
   a) Multi-select the fish body, fins, tail and eyeballs and press **CTRL+G** to group them.
   b) Rename the group *Big fish*.



Creating another fish from a duplicate

1. in the World Explorer, **ALT**-click the Big fish group icon.
2. All the objects in the group are selected.
3. Press **CTRL+D** to duplicate the objects.

   The duplicated objects that make up the fish appear as selected in the World Explorer.
4. Group the selected objects and call the new group *Bottom feeder.*
5. Translate Bottom feeder group downward by five units.
6. Scale it roughly to one third the size of Big fish.
7. Expand Bottom feeder and select Fish body1, then choose Visualization > Add Material to give the fish body a different color.
8. Save your work as *My fish.axel*.

12 Keying, Recording, Preset Animation
Fish — Animating

Now you can animate the Big Fish in the middle of the Aquarium, blowing bubbles, as the Bottom Feeder darts around in deeper waters.

In this part of the tutorial, you use three animation techniques to bring two fish to life:

Keying — For any parameter, you set key values at specific times. AXEL calculates the parameter values in-between the keys resulting in a continuous transition from one key to another. When keying, you turn off the Play button.

Preset Animation — AXEL creates start and end keys automatically.

Record Animation — AXEL records parameter values as you change them interactively during playback. When recording animation, you turn on the Play button.

Need help? — Refer to the *Fish_anim_complete.axel* file in the AXEL / Documentation / Tutorials folder.

In this lesson:

Making the big fish breathe (keying)

To animate the shape of the fish's mouth as it changes over time, we will use shape animation. Shape animation tracks the changing of point position in geometry. Instead of animating the translation, rotation and scaling of an object, we are animating the position of specific points.

To create our shape animation, we will use *keying*. This method gives you the most control over the initial results of your animation. By going to a specific time, and recording a specific value for whichever parameter you are animating, you create *key* positions, and AXEL will interpolate between them for you.

1. Open the *My fish.axel* file if you completed the previous tutorial or open the *Fish_modeling_completed.axel* file.
   An AXEL file displays a World containing two fish groups.

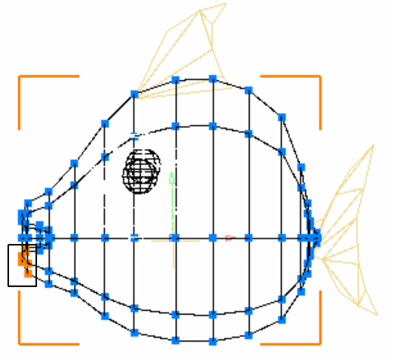
2. Turn off the visibility of the Bottom feeder group.

3. Create an animation sequence for the shape of the fish.

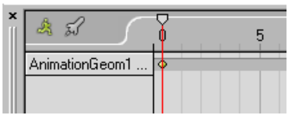
   a) Frame the Fish body by selecting it and pressing **SHIFT + F**.

   b) Switch to Point Mode (press **4**).

   c) Zoom in to the mouth area and select points on the lower lip.



   d) Turn off Play, and go to time 0.



   e) Turn on the Translate button, and choose Animation > Add Animation Sequence.
   In the Sequencer, notice the new sequence with a key at time zero. AXEL has saved the position of all the points on the fish body at time 0.

4. Add two more keys with the mouth at rest:

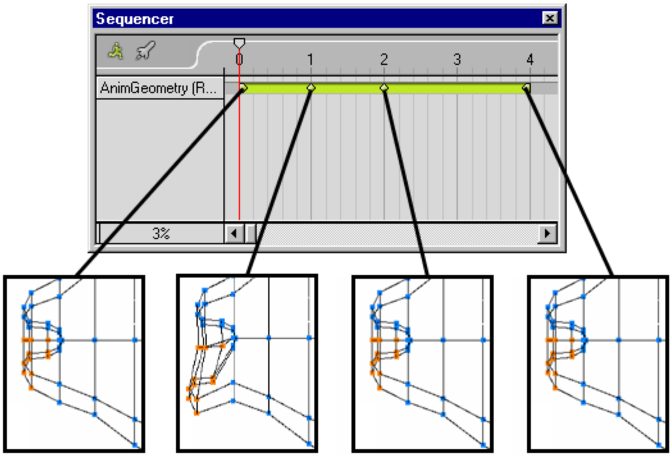
   a) Go to time 2 and press **CTRL+K**.
   This adds a key at this second, so the mouth will return to its original position.

   b) Go to time 4.0 and press **CTRL+K**.
   Adding a key at 4 seconds will give our fish a little pause in his breathing, before the cycling makes him move his lip again.

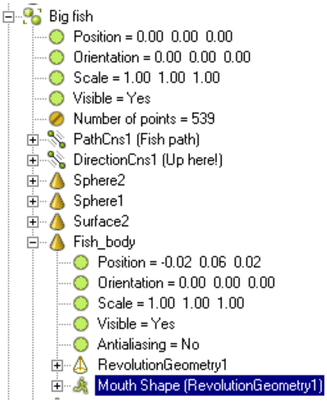
5. Add one key with the mouth open:

   c) Go to time to 1.

   d) Translate the points of the bottom lip so that the fish's mouth opens.
   When you move the points, a key is automatically created. Your animation should have the following mouth positions:



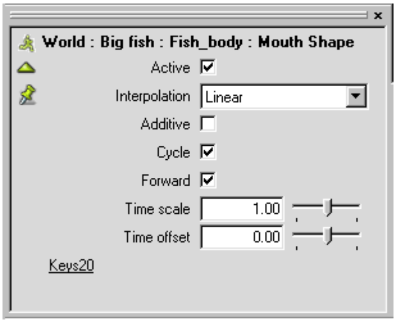
6. In the World Explorer, expand the fish body under the Big Fish group and rename the shape animation Mouth Shape.



```
Big fish
    Position = 0.00  0.00  0.00
    Orientation = 0.00  0.00  0.00
    Scale = 1.00  1.00  1.00
    Visible = Yes
    Number of points = 539
    PathCns1 (Fish path)
    DirectionCns1 (Up here!)
    Sphere2
    Sphere1
    Surface2
    Fish_body
        Position = -0.02  0.06  0.02
        Orientation = 0.00  0.00  0.00
        Scale = 1.00  1.00  1.00
        Visible = Yes
        Antialiasing = No
        RevolutionGeometry1
        Mouth Shape (RevolutionGeometry1)
```

7. Select the Mouth Shape animation and in the Parameter Editor, turn on Cycle.

   The fish will now breathe on a 4 second cycle, using the shape animation you keyed.



World : Big fish : Fish_body : Mouth Shape

Active ☑

Interpolation Linear

Additive □

Cycle ☑

Forward ☑

Time scale 1.00

Time offset 0.00

Keys20

Animating the big fish on a path (preset animation)

Create a ten-second animation of the Big fish travelling along a path. To do so, use a preset animation technique in which AXEL automatically creates a keyframe of the fish position parameter at the start of the path (second 0) and a second key at the end of the path (second 10).

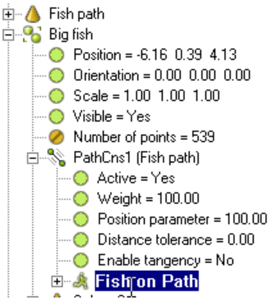
1. Draw a path for the big fish to follow.
   a) Choose Modeling > Draw Curve, or hold L, and draw a path in the top viewport.
   b) Press **END** to close the path.
   c) Rename the curve *Fish path*.

2. Animate the fish along the path:

   a) Multi-select Big fish, then Fish path, in that order.

   b) Choose Animation > Path Animation.

   c) In the World Explorer, expand Big fish / PathCns1 and rename AnimationParam *Fish on path*.

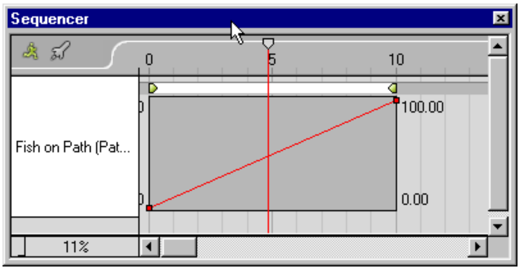


Fish path
Big fish
Position = -6.16 0.39 4.13
Orientation = 0.00 0.00 0.00
Scale = 1.00 1.00 1.00
Visible = Yes
Number of points = 539
PathCns1 (Fish path)
Active = Yes
Weight = 100.00
Position parameter = 100.00
Distance tolerance = 0.00
Enable tangency = No
Fish on Path

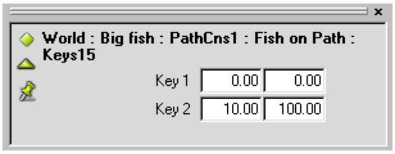
If you drag the playhead in the Sequencer, you will see how the fish is constrained to the path. Switch to Browse mode to play the animation. Switch back to Author mode to continue working on the animation.

3. Add keys to the path animation:

   a) In the Sequencer, double click the animation sequence to display the animation curve. The curve plots the position of the fish in percentage of distance travelled (Y axis) over time (X axis).



   b) Adjust key values by dragging the keys in the Sequencer, or by clicking on a key and editing its values in the Parameter Editor.

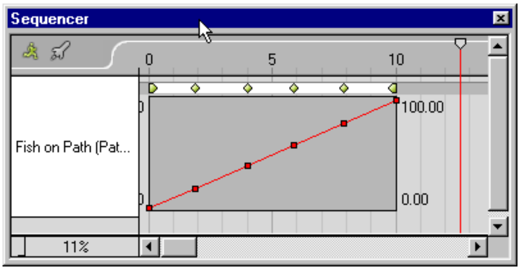


World : Big fish : PathCns1 : Fish on Path : Keys15

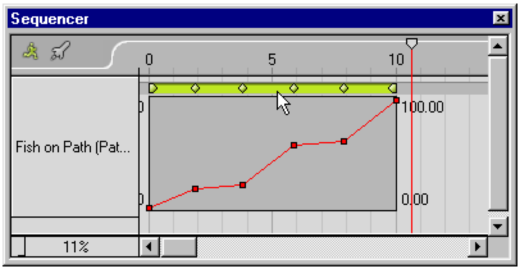
Key 1	0.00	0.00
Key 2	10.00	100.00

c) Go to time 2, select the animation sequence, and press **CRTL+K**.
A key is created at the 2-second mark.

d) Repeat the previous step at times 4, 6 and 8.



e) Click and drag the points on the animation curve until it resembles the illustration below.

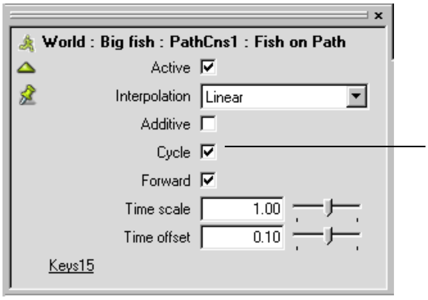


)

The curve indicates that the fish has traveled 20% of the total distance by the time it reaches the second key at time 2. It then remains almost immobile for the next two seconds. It then covers 40% of the total distance between the third and fourth key. It rests once again at the fifth key before completing the last 40% of its journey at the sixth and last key.

f) Switch to Browse mode to preview the animation. Switch back to author mode and experiment with the fish animation by tweaking the animation curve.

g) Select the animation sequence and, in the Parameter Editor, turn on Cycle.



World : Big fish : PathCns1 : Fish on Path

Active ☑

Interpolation Linear

Additive □

Cycle ☑

Forward ☑

Time scale 1.00

Time offset 0.10

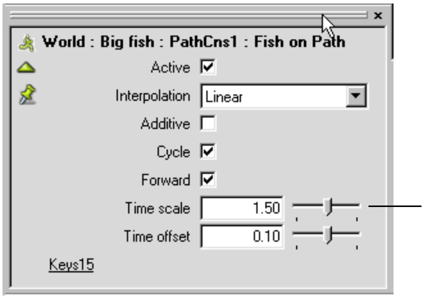
Keys15

Turn Cycle on

The ten-second animation will now play in a continuous cycle.

4. To speed up or slow down the entire animation without moving the keys themselves, you can adjust the sequence's timing:

   a) In the Parameter Editor, adjust the Time scale to 1.50.



World : Big fish : PathCns1 : Fish on Path

Active ☑

Interpolation Linear

Additive □

Cycle ☑

Forward ☑

Time scale 1.50

Time offset 0.10

Keys15

Drag Time scale to 1.5

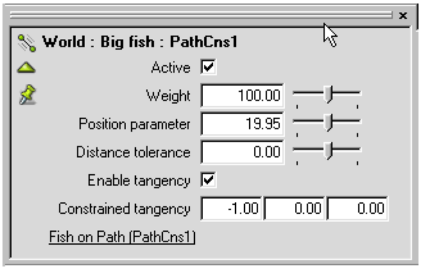
   b) Play the animation.
   The fish now travels along its path at one and a half times the speed it did before.

5. Notice the fish is always looking in the same direction while traveling along the path. It is not very realistic movement. Use tangency to force the Big Fish group to turn and swerve along the path.

   a) In the World Explorer, select the Big fish group. In the Parameter Editor, click PathCns1 to expand it.

   b) In the Parameter Editor, turn on Enable tangency.

   c) Set Constrained tangency to -1.0 0.0 0.0. Now the negative X-axis of the Big Fish group will always be tangent to the path.



World : Big fish : PathCns1

Active ☑

Weight 100.00

Position parameter 19.95

Distance tolerance 0.00

Enable tangency ☑

Constrained tangency -1.00 0.00 0.00

Fish on Path (PathCns1)

   d) Drag the playhead from 0 to 10.
   The fish now follows the curve. Since the curve is swerving in all directions, the fish will sometimes tips over.

6. Prevent the fish from tipping by adding a direction constraint:

   a) Get a cube and rename it *Up here!*

b) Translate it up in the Y axis by 15 units.

c) Press **A** in Front view to see all the objects.

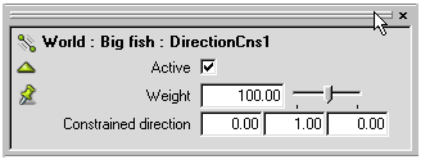
d) In the World Explorer, multi-select the Big Fish group, and then the Up here! cube, in that order.

e) Choose Relations > Direction Constraint.

f) Press **D** in the Front view to display constraints

g) By default, the Direction constraint uses the X-axis of an object. Correct this by selecting the Big Fish group in the World Explorer, and clicking on DirectionCns (Up here!).

h) Set the Constrained direction to 0.0 1.0 0.0 so that the Y-axis is pointing towards the Up here! cube.



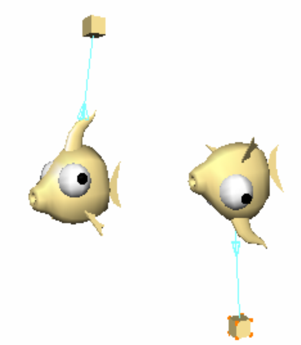
World : Big fish : DirectionCns1

Active ☑

Weight 100.00

Constrained direction 0.00 1.00 0.00

Note: Translating the Up here cube up and down along the Y-axis, will affect which way the Big Fish group's Y-axis will point. You can animate it downwards to make the fish go belly up!

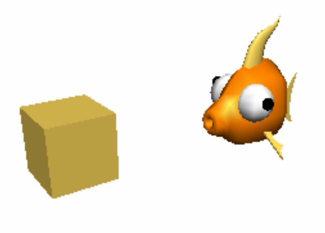


7. Select the Up here! cube and turn off its visibility.

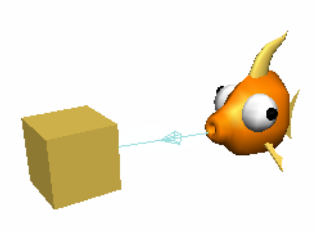
Animating bottom feeder in real time (recording animation)

Now, animate the Bottom Feeder. While playing the animation, turn on the Record button and drag the fish around. AXEL will capture the fish's position and save five keys every second.

1. In the World Explorer, select the Big Fish group and toggle off its visibility. Keep the Fish path visible to use as a reference, since it will help determine the size of the fishbowl.
2. Select the Bottom feeder group and turn on its visibility.
3. Control the fish's direction by constraining it to a cube.
   a) Get a cube and rename it *Chase me!*
   b) Translate the cube downwards and to the left so that Bottom feeder is looking at it.



   c) In the World Explorer, multi-select the Bottom Feeder group and the Chase me! cube, in that order.
   d) Unlock the Lock button, then choose Relations > Direction Constraint.
   By default, the direction constraint would have aligned the positive X-axis of the group to point towards the cube. By unlocking constraints, we created an offset and accepted the direction constraint as is, with the fish looking at the cube.
   e) Turn on the Lock button.
   f) Choose View > Display > Constraints (**SHIFT+D**) to view the constraints in all windows.

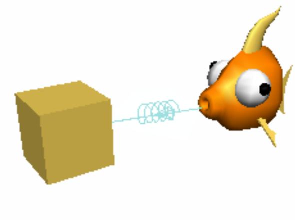


4. Use the cube to drag the fish around. Attach the fish to the cube with a spring, so that its motion is a little unpredictable.

   a) In the World Explorer, multi-select the Bottom Feeder group and the Chase me! cube, in that order.

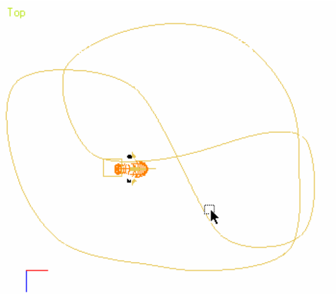
   b) Choose Relations > Spring Constraint
   The spring constraint will allow some liberty to the fish when we drag the cube along, but will always try to force it back, generating some random movement. Play must be on for the spring constraint to take effect. The Direction constraint will force the fish to look at the cube, so it is not always staring out in space.

   c) Select the Chase me! cube and translate it around in the top view. Notice the Bottom Feeder trying to catch up and always looking in the cube's direction. We will animate the cube's position and then hide it, to make Bottom Feeder move around!



5. Drag the fish and record the motion. Animate the cube's position in Top view and use the Fish path as an approximate boundary so that the two fish swim in the same area. This will be helpful when you try to fit them in a fishbowl.

   a) In the top window, zoom out so that the Fish path roughly fills up the viewport.



   b) With the Chase me! cube selected, turn on the Translate button.

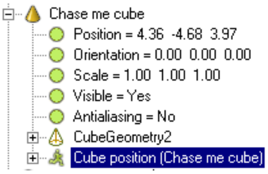
   c) Turn off Play and go to time 0. Turn on the Record button.

   d) When you are ready, start Play. With time running, translate the cube in Top view for approximately 10 seconds. Turn off Record.

AXEL recorded the cube's position by saving one key every 0.2 seconds to track your movement. If your cube was not moving, it will not save additional keys. Switch to Browse mode to preview the result.

If you are not satisfied, you can select the animation track and delete it. Then repeat steps 12 and 13.

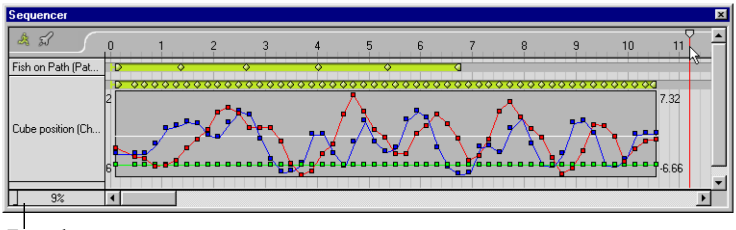
e) In the World Explorer, expand the Chase me! cube and rename AnimationPos "Cube position"



Chase me cube
- Position = 4.36 -4.68 3.97
- Orientation = 0.00 0.00 0.00
- Scale = 1.00 1.00 1.00
- Visible = Yes
- Antialiasing = No
- CubeGeometry2
- Cube position (Chase me cube)

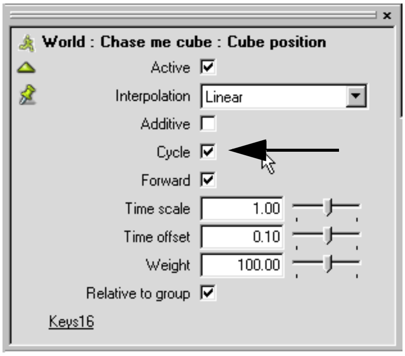
6. Adjust the recorded animation

a) In the Sequencer, double-click the Cube position animation sequence to display the animation curve. To zoom in, drag the zoom bar at the bottom left of the sequencer.



Zoom bar

b) With the sequence selected, turn on cycle in the Parameter Editor.



World : Chase me cube : Cube position

Active ☑

Interpolation Linear

Additive ☐

Cycle ☑

Forward ☑

Time scale 1.00

Time offset 0.10

Weight 100.00

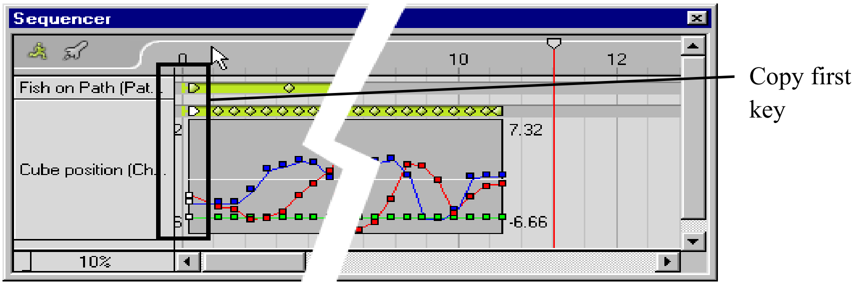
Relative to group ☑

Keys16

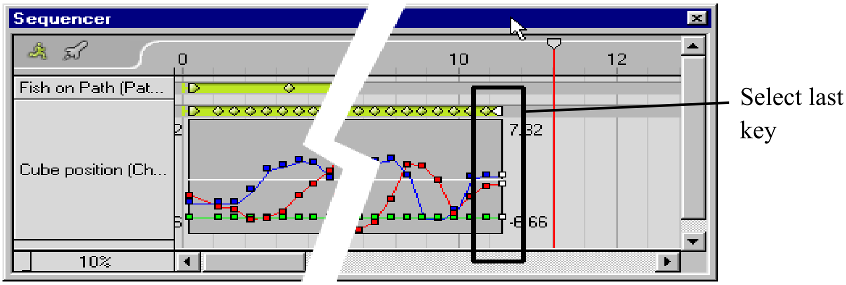
Since the animation is cycled, the fish will jump when it returns from the last key to the first key, unless they both have the same values. Use the copy/paste function on keys to fix this.

c) In the Sequencer, select the first key of the Cube position animation sequence, and press **CTRL+C**.

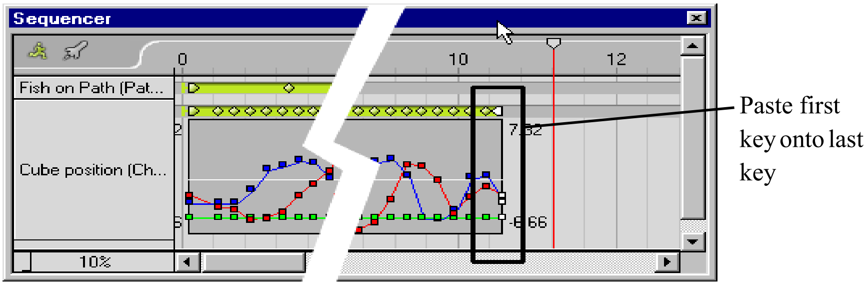
d) Select the last key of the sequence and press **CTRL+V**.
You have now replaced the XYZ position values of the last key with those of the first. This will ensure a smoother transition in the cycle. Refer to the illustration below to notice the difference in values in the curve as you copy/paste keys.



Copy first key



Select last key



Paste first key onto last key

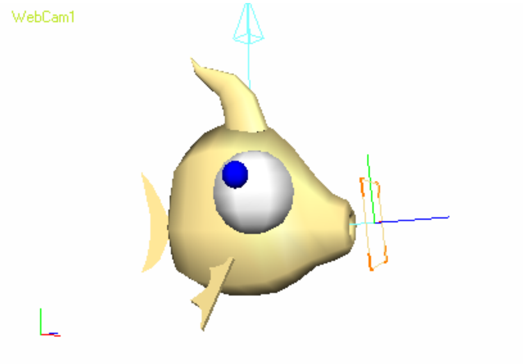
7. Select the Chase me! cube and toggle off its visibility, and switch to Browse mode to preview the animation.

   The Bottom feeder should now be swimming around on a loop.

Creating bubbles with particles (preset animation and keying)

Now, animate bubbles coming out of the mouth of the Big Fish, using Particles. By animating specific parameters, you will be able to control the amount of particles flowing out. You will also add a gurgling sound to accompany the bubbles.

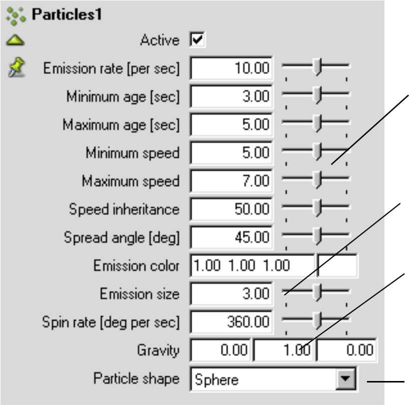
1. Use a rectangle as a particle emitter:
   a) In the World Explorer, select the Big fish group and turn on its visibility.
   b) Choose Modeling > Preset Curves > Square.
   c) Translate it so that it is right in front of the mouth of the Big fish



   d) Rename the square Bubble dummy.
   This square will serve as the object which will gurgle out bubbles. Since the actual geometry of the object has no bearing on the particles, you can use a very low polygon object.
   e) Multi-select the Bubble dummy square and the fish body, in that order.
   f) Unlock the Lock button to allow an offset. Choose Relations > Pose Constraint. Turn on the Lock button.
   The square will now always be in front of the mouth.

2. Add a particle animation to the square:
   a) Select the Bubble dummy square and choose Animation > Add Particles.
   To see the particles, make sure the Play button is on.
   b) With the Bubble dummy square selected, turn off its visibility.

c) In the Parameter Editor, click Particles and modify the parameters as follows:



Particles1

Active ☑

Emission rate [per sec] 10.00

Minimum age [sec] 3.00

Maximum age [sec] 5.00

Minimum speed 5.00

Maximum speed 7.00

Speed inheritance 50.00

Spread angle [deg] 45.00

Emission color 1.00 1.00 1.00

Emission size 3.00

Spin rate [deg per sec] 360.00

Gravity 0.00 1.00 0.00

Particle shape Sphere

Adjust Min. and Max Speed to make the particles flow faster.

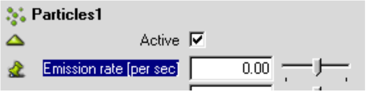
Increase the Emission size for bigger particles.

Change the gravity to 0, 1, 0 so the bubbles flow up.

Select the Sphere shape.

3. Animate the particle emission rate. The Emission rate controls the number of particles flowing out per second. By animating this parameter, we will control the bubbles coming out of the mouth of Big fish.

a) Set the Emission rate to zero.



Particles1

Active ☑

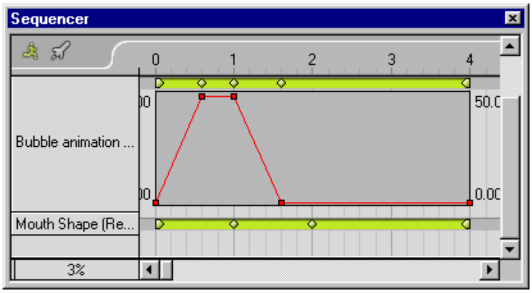
Emission rate [per sec] 0.00

b) Make sure Play is turned off and go to time 0.

c) With the Emission rate at zero turn on the Record button.
An animation sequence is added to the Sequencer.

d) We will coordinate the bubbles with the mouth shape animation. Key the Emission rate with the following values:

0.0	0
0.6	50
1.0	50
1.6	0
4.0	0

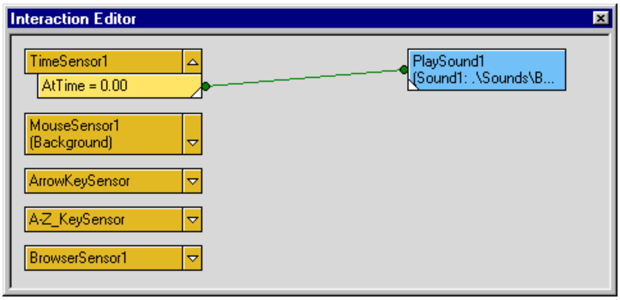


e) Select the animation sequence and turn on Cycle in the Parameter Editor.

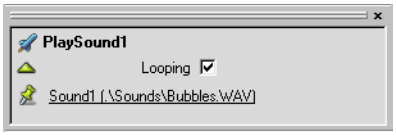
Adding sound

As a final touch, we will cycle a sound file to accompany our bubbles.

1. Choose Interaction > Reaction > Play Sound.
2. Select the Bubbles.wav from the Sounds folder
3. In the Interaction Editor, expand Time Sensor.
4. Link AtTime = 0.00 with the Play Sound reaction.



5. Select the Play Sound reaction and in the Parameter Editor, turn on Looping



PlaySound1

Looping ☑

Sound1 (.\Sounds\Bubbles.WAV)

6. Save your file as *My animated fish.axel*

13 Publishing and Exporting Parameters
Fish — Publishing

In this lesson, you use the publishing tools to incorporate a 3D fish into a 2D web page.

You also learn how to export the fish color parameter so that surfers can modify the parameter on the web page. You use JavaScript to change the color of the fish according to the temperature of the water. When the surfer's mouse rolls over the thermometer, the temperature changes and the fish changes color.

For more information on Web Integration, refer to Web Integration in AXEL On-line help.

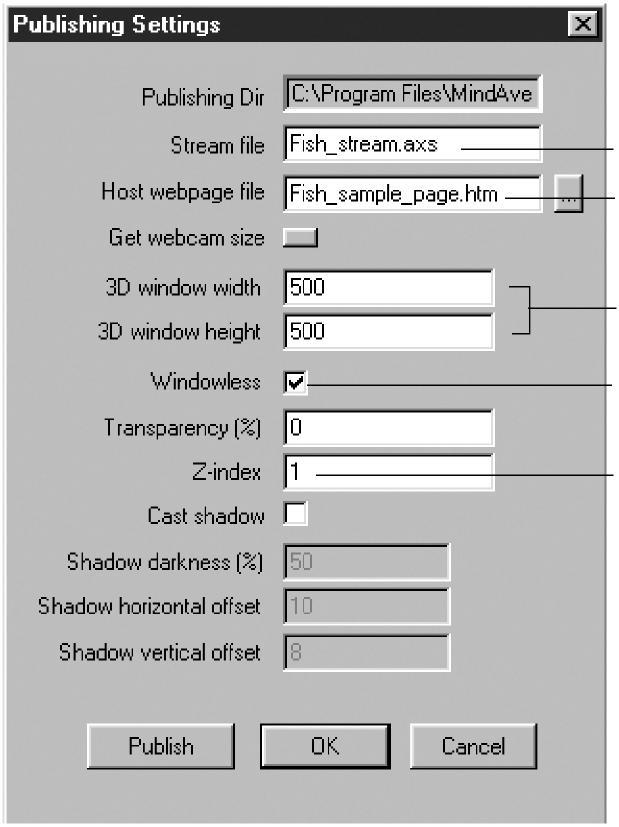
In this lesson:

Publishing the fish in a sample web page

Publishing creates two files: a stream file, and an HTML file in which the stream file is embedded. Here you publish the fish project and make the content windowless so that you can see the 2D background behind the fish.

1. Open *My animated fish.axel* if you completed the previous tutorial or open *Fish_anim_completed.axel*.
   An AXEL file displays a world containing the two animated fish groups. Although you will hide the Fish curve, it will be useful to determine the final camera position.

2. Choose Web Integration > Publishing Settings, enter the following information, and click Publish.



Publishing Settings

Publishing Dir	C:\Program Files\MindAve
Stream file	Fish_stream.axs
Host webpage file	Fish_sample_page.htm
Get webcam size	
3D window width	500
3D window height	500
Windowless	☑
Transparency (%)	0
Z-index	1
Cast shadow	□
Shadow darkness (%)	50
Shadow horizontal offset	10
Shadow vertical offset	8

Names the published stream file

Names HTML file that will host the published stream file

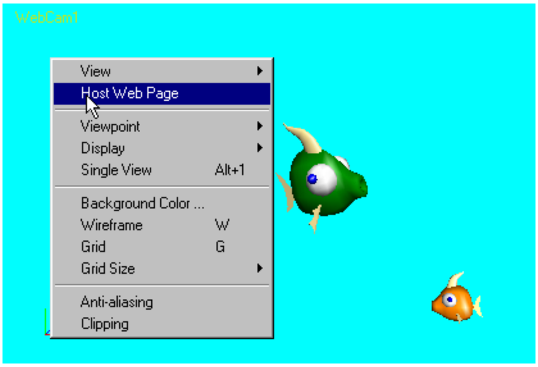
Sets the size of 3D window in pixels

Publishes the fish without a background

Layers the fish in front of the 2D content on the web page

The stream file *fish.axs* and the host web page *fish_sample.htm* are stored in your project directory.

3. To preview the page, right-click in the WebCam viewport and choose Host Web page.



The sample web page is displayed in the viewport.

4. Choose Web Integration > Download Time Estimates to see the time the AXEL content will take to download. Compare various surfer connection speeds to get an idea of what they can expect.

   By default, the geometry is not streamed. The file will begin playing when all geometry, animation and interaction is loaded. The sound files, usually heavier, are considered high-priority streamed files, which means that the file will start playing without the bubbling sound, which will start playing as soon as it is downloaded. The same is true for textures.

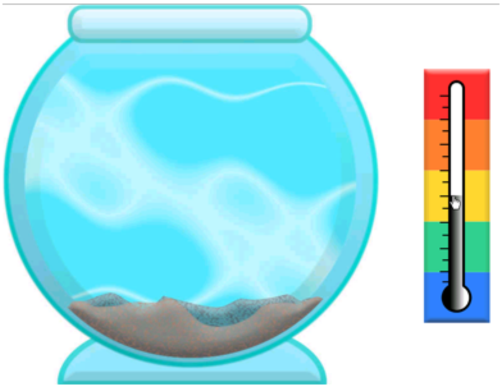
Publishing the fish in an existing web page

Now that you have tested the published content in a sample page, it can be integrated in a more elaborate web page design. Use the HTML page that was created for this lesson.

Note: The fish_bowl.htm web page was designed in a third-party web editor. This code is provided as an example of how you can integrate JavaScript, HTML, and AXEL content.

1. Preview the web page that will host the published fish:
   a) Launch a web browser.

b) Open Fishbowl.htm in the *Documentation \ Tutorials \ Fish* folder.
   Notice how the mercury rises when you move your mouse over the thermometer.



2. Examine the HTML code:

   a) Open the file in an HTML editor and examine the source.
      Notice the script used to create the thermometer rollover. We will use it later on in this lesson to control parameters in the AXEL stream file.

The first part of the code defines the functions used for the thermometer rollover.

```
<html>
<head>
<title>Fishbowl</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

<script>
<!--
//-->

function MM_preloadImages() { //v3.0
  var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
    var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
    if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
}

function MM_swapImgRestore() { //v3.0
  var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
}

function MM_findObj(n, d) { //v4.0
  var p,i,x;  if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
    d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
  if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
  if(!x && document.getElementById) x=document.getElementById(n); return x;
}

function MM_swapImage() { //v3.0
  var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array; for(i=0;i<(a.length-
2);i+=3)
   if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x; if(!x.oSrc) x.oSrc=x.src; x.src=a[i+2];}
}
//-->
</script>
```

b) The rest of the code defines the body of the page. Notice the two tables: the first table contains one row, with two columns. The first column contains the fish bowl image. The second column contains another table defining the thermometer. Rollovers are used to swap the thermometer images so that it changes color when the mouse rolls over it.

```
<body bgcolor="#FFFFFF" onLoad="MM_preloadImages('images/ther_r1_c1_f5.gif','images/ther_r2_c1_f5.gif','imag-
es/ther_r3_c1_f5.gif','images/ther_r4_c1_f5.gif','images/ther_r5_c1_f5.gif','images/ther_r1_c1_f4.gif','images/
ther_r2_c1_f4.gif','images/ther_r3_c1_f4.gif','images/ther_r4_c1_f4.gif','images/ther_r5_c1_f4.gif','images/
ther_r1_c1.gif','images/ther_r2_c1.gif','images/ther_r3_c1.gif','images/ther_r4_c1.gif','images/ther_r5_c1.gif','images/
ther_r1_c1_f3.gif','images/ther_r2_c1_f3.gif','images/ther_r3_c1_f3.gif','images/ther_r4_c1_f3.gif','images/
ther_r5_c1_f3.gif','images/ther_r1_c1_f2.gif','images/ther_r2_c1_f2.gif','images/ther_r3_c1_f2.gif','images/
ther_r4_c1_f2.gif','images/ther_r5_c1_f2.gif')">
<table width="500" border="0" cellspacing="10" cellpadding="30" align="left" bgcolor="#FFFFFF">
  <tr>
    <td>
      <div align="left"><img src="images/fishbowl1.jpg" width="587" height="598"></div>
    </td>
```

Fish bowl image

```
    <td>
      <div align="center">
        <table border="0" cellpadding="0" cellspacing="0" width="103" align="center">
        <!-- fwtable fwsrc="thermometer4.png" fwbase="ther.gif" fwstyle="Dreamweaver" fwdocid = "742308039"
fwnested="0" -->
          <tr>
            <td><img src="images/spacer.gif" width="103" height="1" border="0" name="undefined_2"></td>
            <td><img src="images/spacer.gif" width="1" height="1" border="0" name="undefined_2"></td>
          </tr>
          <tr>
            <td><a href="#" onMouseOver="MM_swapImage('ther_r1_c1','','images/
ther_r1_c1_f5.gif','ther_r2_c1','','images/ther_r2_c1_f5.gif','ther_r3_c1','','images/
ther_r3_c1_f5.gif','ther_r4_c1','','images/ther_r4_c1_f5.gif','ther_r5_c1','','images/ther_r5_c1_f5.gif',1);" ><img
name="ther_r1_c1" src="images/ther_r1_c1.gif" width="103" height="78" border="0"></a></td>
            <td><img src="images/spacer.gif" width="1" height="78" border="0" name="undefined_2"></td>
          </tr>
          <tr>
            <td><a href="#" onMouseOver="MM_swapImage('ther_r1_c1','','images/
ther_r1_c1_f4.gif','ther_r2_c1','','images/ther_r2_c1_f4.gif','ther_r3_c1','','images/
ther_r3_c1_f4.gif','ther_r4_c1','','images/ther_r4_c1_f4.gif','ther_r5_c1','','images/ther_r5_c1_f4.gif',1)" ><img
name="ther_r2_c1" src="images/ther_r2_c1.gif" width="103" height="79" border="0"></a></td>
            <td><img src="images/spacer.gif" width="1" height="79" border="0" name="undefined_2"></td>
          </tr>
          <tr>
           <td><a href="#" onMouseOver="MM_swapImage('ther_r1_c1','','images/ther_r1_c1.gif','ther_r2_c1','','im-
ages/ther_r2_c1.gif','ther_r3_c1','','images/ther_r3_c1.gif','ther_r4_c1','','images/ther_r4_c1.gif','ther_r5_c1','','imag-
es/ther_r5_c1.gif',1)" ><img name="ther_r3_c1" src="images/ther_r3_c1.gif" width="103" height="80"
border="0"></a></td>
            <td><img src="images/spacer.gif" width="1" height="80" border="0" name="undefined_2"></td>
          </tr>
          <tr>
            <td><a href="#" onMouseOver="MM_swapImage('ther_r1_c1','','images/
ther_r1_c1_f3.gif','ther_r2_c1','','images/ther_r2_c1_f3.gif','ther_r3_c1','','images/
ther_r3_c1_f3.gif','ther_r4_c1','','images/ther_r4_c1_f3.gif','ther_r5_c1','','images/ther_r5_c1_f3.gif',1)" ><img
name="ther_r4_c1" src="images/ther_r4_c1.gif" width="103" height="79" border="0"></a></td>
            <td><img src="images/spacer.gif" width="1" height="79" border="0" name="undefined_2"></td>
          </tr>
          <tr>
            <td><a href="#" onMouseOver="MM_swapImage('ther_r1_c1','','images/
ther_r1_c1_f2.gif','ther_r2_c1','','images/ther_r2_c1_f2.gif','ther_r3_c1','','images/
ther_r3_c1_f2.gif','ther_r4_c1','','images/ther_r4_c1_f2.gif','ther_r5_c1','','images/ther_r5_c1_f2.gif',1)" ><img
name="ther_r5_c1" src="images/ther_r5_c1.gif" width="103" height="80" border="0"></a></td>
            <td><img src="images/spacer.gif" width="1" height="80" border="0" name="undefined_2"></td>
          </tr>
        </table>
```

Thermometer

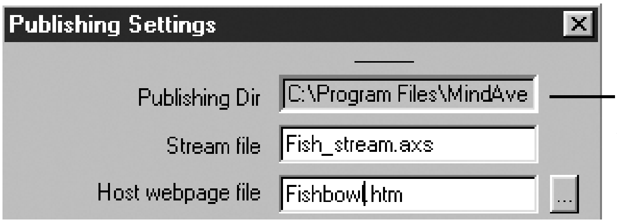
```
</div>
    </td>
  </tr>
</table>
</body>
</html>
```

3. Place the web page and the AXEL project file into the same project directory:

   a) In AXEL, choose File > Save As, and save the fish project in *Documentation \ Tutorials \ Fish*.

      Now the project file, the web page, and its associated image folders are in the same location. If the AXEL project referred to textures or sound, they would automatically be saved in the same project directory. This directory will also be used as the publishing directory.

4. Choose Web Integration > Publishing Settings, change the name of the host web page to *fishbowl.htm* and click Publish.



Publishing Settings

Publishing Dir: C:\Program Files\MindAve

Stream file: Fish_stream.axs

Host webpage file: Fishbowl.htm

Notice the publishing directory is now *Documentation \ Tutorials \ Fish.*

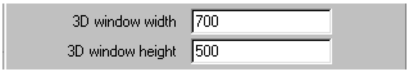
5. When prompted to add the AXEL object to the end of the page, click OK.

   The stream file is saved in the project directory, and embedded in the host web page. In the host web page view, the fish appears in the fish bowl.

   The rectangular outline around the fish indicates the size of the 3D window.

6. Notice the 3D window is to the left of the web page. When you play the animation, the fish swims outside the bowl. Adjust the 3D content by publishing a new stream file:

   a) Choose Web Integration > Publishing Settings, and set the 3D Window Width and Height to 700 and 500 respectively, and click Publish.



3D window width: 700

3D window height: 500

      Now the 3D window is bigger, and its center is approximately in the center of the fish bowl.

   b) Turn on the visibility of the Fish path to see where the fish swims.

   c) Zoom, pan and orbit in WebCam view until the fish stays inside the bowl.

   d) Switch to Browse mode to preview the result.

   e) When the fish is in the right position, turn off the visibility of the Fish path.

   f) Choose Web Integration > Publish.

7. Edit the HTML code to change the layer position of the 3D window in the host web page.
   a) Choose Web Integration > View Source Host Web Page.
   The Fishbowl.htm file is displayed in a text editor. Notice the embedded AXEL object at the end of the page.
   b) In the style attribute, set top to 100, and save the file.

```
<!-- Begin AXEL object -->

<object id="Fish_stream" classid="CLSID:68A2C3BD-7809-11D3-
8ACF-0050046F2F9A"
   codebase="http://www.mindavenue.com/Downloads/
AXELPlayerAX_Win32.cab#version=1,0,793,0"
   style="position:absolute; left:5; top:100; z-index:1; filter:alpha(opaci-
ty=100)"
   width="700" height="500">
  <param name="Src" value="Fish_stream.axs">
  <param name="Windowless" value="1">
  <embed src="Fish_stream.axs" name="Fish_stream"
    pluginspage="http://www.mindavenue.com/Downloads/AxelPlayerN-
PInstall.htm"
     type="application/x-MindAvenueAxelStream"
     style="position:absolute; left:5; top:100; z-index:1;"
     width=700 height=500 Windowless=true>
  </embed>
```

   c) Open the fishbowl.htm in a web browser.
   Now the bottom feeder is closer to the sand.

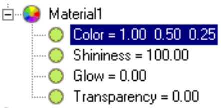
Note: To add 3D content to the existing fishbowl web page, you could also cut and paste the embedded AXEL object from the sample page into the fish bowl page. You could place it at the end of the page, or next to the fishbowl image in the row-1, column-1 of the first table.

Exporting a color parameter

Add color to the big fish, then export the color parameter so that it can be controlled in the web page.

1. In AXEL, select the fish body of the Big fish and choose Visualization > Add material. Select a color.

2. In the World Explorer, expand the new material under Big fish / Fish body, and select its color parameter.

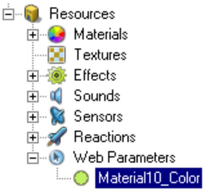


Material1
Color = 1.00 0.50 0.25
Shininess = 100.00
Glow = 0.00
Transparency = 0.00

Note: You can also select the parameter in the Parameter Editor.

3. Choose Web Integration > Export Parameter.

   The color parameter is added to the parameters to be exported next time you publish the fish. It is listed in the World Explorer \ Resources \ Web Parameters.



4. Choose Web Integration > Publish.

5. To preview the result, choose Web Integration > Preview in IE. Click the parameter buttons to change the color of the fish.

   Note: In v1.0, the parameter buttons do not work with windowless content in Host Web Page view.

6. In the World Explorer, expand Resources \ Web Parameters, click Material_Color twice and enter *fish_color*.

7. Choose Web Integration > Publish.

   The button name is now *fish_color*.

Linking the exported parameter to a JavaScript rollover

Use the thermometer to control the color of the fish, then remove the default grey buttons.

Note: For an example of the complete HTML code, refer to fishbowl_complete.htm in the Documentation \ Tutorials \ Fish directory.

1. View the JavaScript that was generated when you exported the color parameter. In AXEL, choose View Source Host Web Page.
   The sample code was added after the embedded object tag at the end of the HTML file.

```
<!--AxelBeginWebParamCode-- This code was automatically generated by Axel -
changes in this section may be overwritten! -->

<div style="position:absolute; left:5; top:5; z-index:0;">

<script language="JavaScript">
  function setfish_color( i_NewValue )
  {
    document.Fish_stream.SetWebParam( "fish_color", i_NewValue );
  }
  function getfish_color( i_NewValue )
  {
    return document.Fish_stream.GetWebParam( "fish_color" );
  }
</script>

<form>
<p>fish_color:
  <input type="button" value="Default" OnClick="setfish_color( '255  0  128' )">
   <input type="button" value="Red" OnClick="setfish_color( '1. 0. 0. 1.' )">
   <input type="button" value="Green" OnClick="setfish_color( '0. 1. 0. 1.' )">
   <input type="button" value="Blue" OnClick="setfish_color( '0. 0. 1. 1.' )">
   <input type="button" value="Prompt" OnClick="setfish_color( prompt( 'Change
fish_color value:', getfish_color() ))">
</p>
</form>
</div>

<!-- End of Axel automatically generated code --AxelEndWebParamCode -->
<!-- End AXEL object -->
</body>
</html>
```

2. Add the following function definitions to the script:

• Add the setfish_color function definition that was provided in the Export Parameter sample code.

```
function setfish_color( i_NewValue ){
  document.Fish_stream.SetWebParam( "fish_color", i_NewValue );
}
```

• Also add a function for each of 5 different colors on the thermometer. The function calls two other functions: the rollover function that changes the thermometer image and the set the fish color function that changes the fish color parameter.

```
function red_clicked(){
                MM_swapImage('ther_r1_c1','','imgs/
ther_r1_c1_f5.gif','ther_r2_c1','','imgs/ther_r2_c1_f5.gif','ther_r3_c1','','imgs/
ther_r3_c1_f5.gif','ther_r4_c1','','imgs/ther_r4_c1_f5.gif','ther_r5_c1','','imgs/
ther_r5_c1_f5.gif',1)
                setfish_color( '1. 0. 0. 1.' )
}
```

Rollover

Set parameter

The following is an example of the complete script:

```
<script>
function MM_preloadImages() { //v3.0
  var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
    var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
    if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
}
function MM_swapImgRestore() { //v3.0
  var i,x,a=document.MM_sr; for(i=0;a&&i>a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
}
function MM_findObj(n, d) { //v4.0
  var p,i,x;  if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
    d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
  if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i>d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
  if(!x && document.getElementById) x=document.getElementById(n); return x;
}
function MM_swapImage() { //v3.0
  var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array; for(i=0;i<(a.length-2);i+=3)
   if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x; if(!x.oSrc) x.oSrc=x.src; x.src=a[i+2];}
}
//-->
function red_clicked(){
                         MM_swapImage('ther_r1_c1','','imgs/ther_r1_c1_f5.gif','ther_r2_c1','','imgs/
ther_r2_c1_f5.gif','ther_r3_c1','','imgs/ther_r3_c1_f5.gif','ther_r4_c1','','imgs/ther_r4_c1_f5.gif','ther_r5_c1','','imgs/
ther_r5_c1_f5.gif',1)
                         setfish_color( '1. 0. 0. 1.' )
}
function orange_clicked(){
                         MM_swapImage('ther_r1_c1','','imgs/ther_r1_c1_f4.gif','ther_r2_c1','','imgs/
ther_r2_c1_f4.gif','ther_r3_c1','','imgs/ther_r3_c1_f4.gif','ther_r4_c1','','imgs/ther_r4_c1_f4.gif','ther_r5_c1','','imgs/
ther_r5_c1_f4.gif',1)
                         setfish_color( '1.0 0.5 0.0' )
}
function yellow_clicked(){
                         MM_swapImage('ther_r1_c1','','imgs/ther_r1_c1.gif','ther_r2_c1','','imgs/
ther_r2_c1.gif','ther_r3_c1','','imgs/ther_r3_c1.gif','ther_r4_c1','','imgs/ther_r4_c1.gif','ther_r5_c1','','imgs/
ther_r5_c1.gif',1)
                         setfish_color( '1. 1. 0. 0.' )
}
function green_clicked(){
                         MM_swapImage('ther_r1_c1','','imgs/ther_r1_c1_f3.gif','ther_r2_c1','','imgs/
ther_r2_c1_f3.gif','ther_r3_c1','','imgs/ther_r3_c1_f3.gif','ther_r4_c1','','imgs/
ther_r4_c1_f3.gif','ther_r5_c1','','imgs/ther_r5_c1_f3.gif',1)
                         setfish_color( '0. 1. 0. 1.' )
}
function blue_clicked(){
                         MM_swapImage('ther_r1_c1','','imgs/ther_r1_c1_f2.gif','ther_r2_c1','','imgs/
ther_r2_c1_f2.gif','ther_r3_c1','','imgs/ther_r3_c1_f2.gif','ther_r4_c1','','imgs/
ther_r4_c1_f2.gif','ther_r5_c1','','imgs/ther_r5_c1_f2.gif',1)
                         setfish_color( '0. 0. 1. 1.' )
}
function setfish_color( i_NewValue ){
   document.Fish_stream.SetWebParam( "fish_color", i_NewValue );
}
</script>
```

d) Edit the thermometer rollovers in the nested table to use the new functions:

```
<table border="0" cellpadding="0" cellspacing="0" width="103">
      <!-- fwtable fwsrc="thermometer4.png" fwbase="ther.gif" fwstyle="Dreamweaver" fwdocid = "742308039" fwnested="0" -->
       <tr>
         <td><img src="imgs/spacer.gif" width="103" height="1" border="0" name="undefined_2"></td>
         <td><img src="imgs/spacer.gif" width="1" height="1" border="0" name="undefined_2"></td>
       </tr>
       <tr>
         <td><a href="#" onMouseOver="red_clicked()"><img name="ther_r1_c1" src="imgs/ther_r1_c1.gif" width="103"
height="78" border="0"></a></td>
         <td><img src="imgs/spacer.gif" width="1" height="78" border="0" name="undefined_2"></td>
       </tr>
       <tr>
         <td><a href="#" onMouseOver="orange_clicked()" ><img name="ther_r2_c1" src="imgs/ther_r2_c1.gif" width="103"
height="79" border="0"></a></td>
         <td><img src="imgs/spacer.gif" width="1" height="79" border="0" name="undefined_2"></td>
       </tr>
       <tr>
         <td><a href="#" onMouseOver="yellow_clicked()" ><img name="ther_r3_c1" src="imgs/ther_r3_c1.gif" width="103"
height="80" border="0"></a></td>
         <td><img src="imgs/spacer.gif" width="1" height="80" border="0" name="undefined_2"></td>
       </tr>
       <tr>
         <td><a href="#" onMouseOver="green_clicked()"><img name="ther_r4_c1" src="imgs/ther_r4_c1.gif" width="103"
height="79" border="0"></a></td>
         <td><img src="imgs/spacer.gif" width="1" height="79" border="0" name="undefined_2"></td>
       </tr>
       <tr>
         <td><a href="#" onMouseOver="blue_clicked()" ><img name="ther_r5_c1" src="imgs/ther_r5_c1.gif" width="103"
height="80" border="0"></a></td>
         <td><img src="imgs/spacer.gif" width="1" height="80" border="0" name="undefined_2"></td>
       </tr>
      </table>
```

e) To remove the sample buttons, delete the form tag and its contents.

```
<form>
<p>fish_color:
   <input type="button" value="Default" OnClick="setfish_color( '255  0  128' )">
   <input type="button" value="Red" OnClick="setfish_color( '1. 0. 0. 1.' )">
   <input type="button" value="Green" OnClick="setfish_color( '0. 1. 0. 1.' )">
   <input type="button" value="Blue" OnClick="setfish_color( '0. 0. 1. 1.' )">
   <input type="button" value="Prompt" OnClick="setfish_color( prompt( 'Change fish_color value:', getfish_color()
))">
</p>
</form>
```

Index