

# Introduction to Low-Polygon 3D Modeling for Games

Using NewTek Lightwave and Adobe Director

## *Overview*

This document describes some basic methods for creating 3D geometry for use in Shockwave3D games using NewTek Lightwave and Adobe Director. It is intended for the novice Lightwave and Director user, but there are tips and tricks sprinkled throughout that may help the more experienced developer.

To follow along, you will need a copy of Lightwave3D installed. You do not need Director for this tutorial, but it does assume that the content is destined for a Shockwave3D environment. Most of the content applies to any sort of low-polygon modeling, however.

## *An Introduction to the Lightwave Interface*

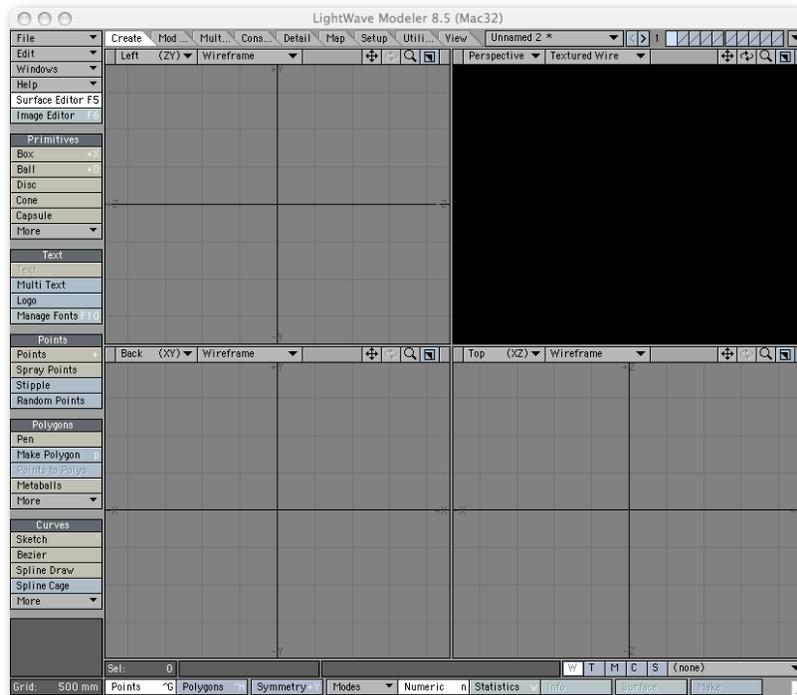
Before we get started, let's familiarize ourselves with the Lightwave interface. Lightwave is a very deep and powerful tool, so we will only be looking at a tiny subset of Lightwave's capabilities, so when you get comfortable with the concepts here, be sure to delve further into the interface to learn more powerful modeling and animation techniques.

### **The Two Core Programs**

Lightwave is not a single program, but a suite of tools that are useful for developing 3D content. We will be focusing on two of them: the 3D modeling program (called "Modeler") and the 3D layout program (called "Lightwave"). Since there is potential confusion between "Lightwave" as the entire package and "Lightwave" the layout program, I'll refer to the layout program as "Layout," even though it is officially called "Lightwave."

### **Introduction to Modeler**

So let's fire up Modeler. This is what Modeler's interface looks like:



Now, Modeler is customizable regarding how it is presented; you can define different screen layouts, etc. If you want to tweak your layouts, or if your preset layout doesn't look like this (say, if you're learning on someone else's workstation), then look into the documentation for how to change the layout. If yours looks similar to the above, then you're ready to work with this tutorial.

## Your View into the 3D World

The first thing you'll notice is that the bulk of the screen is divided into four quadrants. These provide four views into your 3D world. At the top of each quadrant are drop-down menus for selecting the point of view (left, top, etc.) and the rendering style (wireframe, shaded, etc.), and some buttons for navigating the 3D view.

### Viewing your object from different angles

Selecting a view from the first menu will show you the 3D space you'll be creating geometry in from a different perspective; if you select "Left," then you view the object from the left. If you select "Top," then you view the object from above. "Perspective" gives you a free-roaming camera that you can rotate and move around the 3D space to examine your object from any angle.

The button that looks like four arrows (⌕) allows you to pan around your scene. Just click on it and drag to move around. The button that looks like a magnifying glass (🔍) lets you zoom in and out the same way. Viewports that have been set to the "Perspective" view also have a button that looks like a circle of arrows (🔄), which allow you to orbit the camera around the object.

### Viewing your object with different rendering styles

Selecting a rendering type from the second menu will show you the 3D space in different rendering formats ranging from "Wireframe" to "Textured Wire." Different views require different amounts of processor load and show different kinds of details, so it is often useful to switch between modes while working. Here are some of the modes I use the most, from the least processor intensive on the left to the most on the right:



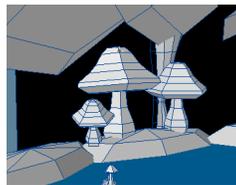
**Wireframe**

This mode shows all points and lines, and lets you "see through" your geometry. It allows the most precise selection of points and surfaces.



**Sketch**

This mode is fast-drawing, but still lets you see what surfaces are in front of others. (Wireframe mode can get confusing at times.)



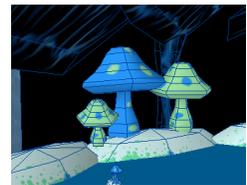
**Wire Shade**

This mode is like sketch, only it includes both shading information (to convey surface facing) and the base color of surfaces.



**Texture**

This mode is the closest to what your object will look like in Shockwave3D, but it can be difficult to understand where the geometry lies.



**Texture Wire**

This mode takes the longest to draw, but it lets you manipulate the geometry easily while still seeing what it will look like in the game.

## Selection Modes

Your geometry will consist of points and surfaces. Points are anchors in space between which the surfaces are stretched. Modeler has two selection modes for selecting each type of geometry information – one for points, and one for surfaces.

The easiest way to toggle between these modes is by pressing the space bar. Try it now. Down at the bottom of the screen, you will see two buttons alternately lighting up:  The highlighted button indicates which mode you are currently in. The cursor also changes between a cross and an “X” to indicate the change, as well. Besides using the space bar, you can also click the buttons, or use the keyboard shortcuts to change the mode (^G and ^H in this case).

Most of the tools you use in Modeler require selection of either points or polygons or both, so if you find that you cannot select what you want to select while working in this tutorial, check your selection mode.

### Adding and subtracting from selections

By default, *everything* is selected. So, if you use the “Move” tool without selecting anything, you’ll move everything. Most of the time, however, you will want to only affect certain points or surfaces. So, you will need to select the ones you want. To do this, simply click on them in any viewport. If you are in Points selection mode, you can select points, and in Polygons selection mode, you can select surfaces. There are some tricky parts, however:

- If you already have a point or polygon selected, clicking acts to *deselect* items. This way, you can select a lot of things and then deselect what you don’t want. However, this can lead to confusion if you have a point or polygon not visible in any of your viewports that is selected, because it will appear that you have no selection. When you click on a point or polygon, it won’t select, because there’s nothing to deselect there. You can clear the current selection at any time by pressing the slash key (“/”) to start a clean selection, and you can always add to a selection by holding the SHIFT key down.
- If you are working in wireframe mode, you cannot click in the middle of a polygon to select it, because your click will go right through it. To select polygons in wireframe, you need to drag over an edge of the polygon. A similar problem is that if you are working in a shaded mode, the surfaces block clicks, so you cannot select a point that is behind a surface.
- If you click a point in a viewport in wireframe mode, and there is another point directly behind it (say you click the corner of a cube in the “Left” view), *both* points will be selected. This may lead to strange results if that was not your intent. To only select the frontmost point, switch to another viewport and deselect the point you don’t want.
- Using the Command key (or the Alt key on Windows) allows you to select using a drawn lasso, useful for selecting large amounts of items.

## Tools

One final thing that we need to cover before we can get started: the tools. Lightwave provides a lot of tools, and they are all accessible at the click of a button. However, there are so many tools that they could not possibly have a button for each tool, so they are organized into tool sets.

The tools available to you in the current set are shown down the lefthand side of the interface. If there is a hotkey defined for the tool, it is shown on the button itself. Some tools (like the File menu, Edit menu, etc.) are always shown, and appear at the top of the lefthand column.

To change sets, click the tabs along the top of the screen. They are labeled “Create,” “Modify,” “Multiply,” etc. according to the logical function of the tools. As you become more familiar with Lightwave, you will learn which tools are found in which tool sets, and as you learn the hotkeys, which can be used regardless of which tool set you have selected, you will have to change tool sets less often. Although you can click on tool buttons to select them, you should make it a habit to use the keyboard shortcut whenever you need to select a tool that has one, because the sooner you internalize the keyboard shortcuts, the faster you will get at modeling your objects.

Most tools either happen instantaneously, or enter a mode where you drag or otherwise define values for an operation, and then you hit return to commit to the operation. Other tools, like the drag tool, just enter a mode where you can continually use the tool to manipulate over and over again. To exit out of a tool, you can usually just press the space bar to jump to a selection mode.



### *Creating an object from a primitive*

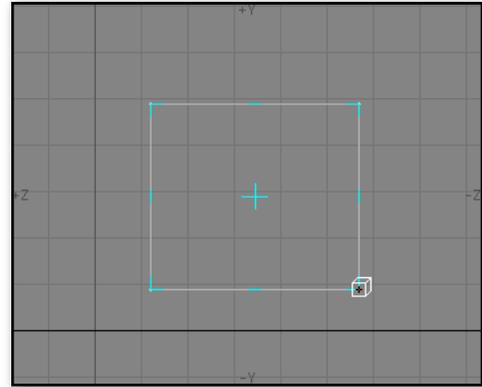
Now that you are familiar with the basic parts of the Modeler, let's create an object. The first thing we will do is create an object from a primitive. Let's create a column, like what you would find on a porch or inside a building to support the ceiling.

Because we're going to be using this column inside a game, we want it to have the lowest polygon count possible, since it must be rendered in real time. Every polygon you spend on an object is a polygon that cannot be used on another object, so conservation is key! Our very first decision plays a profound role on this consideration: is the column round or square?

In the world of Shockwave3D, all surfaces are made up of flat faces. There are no curved surfaces. While we can achieve the illusion of roundedness by using many flat surfaces and some shading tricks, but ultimately, it will take a lot more geometry than a square column. If it is not important to the game world or story for the columns to be round, then we can make them square to conserve polygons, something like the column shown at left. (In fact, even that probably has too much geometry detail to be very useful in a Shockwave3D game. Instead, we will simplify the lines, and add detail with texture maps.

## Creating a box

So let's start by creating a box. Select the "Create" toolset and choose the "Box" tool. When you move your cursor over one of the viewports, you will see it change into a little box. Click and drag out the box in one axis, and you will see a square appear, as shown at right.



This box is currently a flat plane only (and if that's all you wanted, you could hit Return and be done). But to make a box, we need to define its other dimension, so switch over to another viewport and drag again to define the depth.

Position	
X:	77 m
Y:	
Z:	28 m
Grid: 5 m	

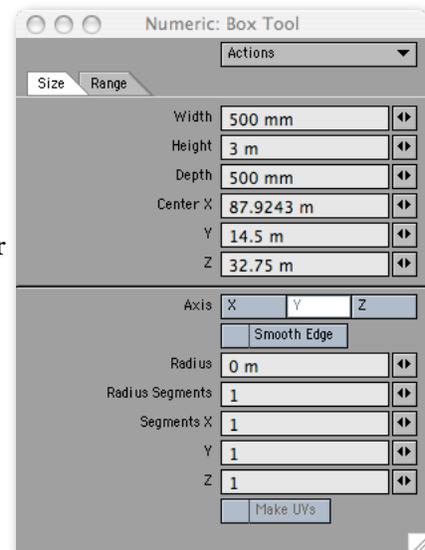
If this doesn't feel very precise to you, it isn't. You can "eyeball" the size if you want, but there are also some tools to help you be more precise. Down in the lower lefthand corner, you will see an information pane that shows you your current location in the 3D world. If nothing else, this gives you a sense of the scale at which you are creating. A 77 meter tall column is pretty huge, so clearly, we need to re-think our scale.

## Precisely defining your primitive

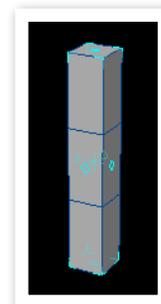
Another approach is to do numeric entry for the size of our box. Press the "n" key to call up the "Numeric: Box Tool" window. Many tools support this behavior; pressing the "n" key while using a tool will call up a window that lets you type in numeric values to define the behavior of the tool.

The numeric entry window allows you to define all aspects of the new box. Let's go in and type in reasonable values for the column:

- **Width:** .5 (because the default unit is meters, this means half a meter - it will automatically change to 500 mm)
- **Height:** 3 (three meters - it will automatically add the units)
- **Depth:** 500mm (same as .5 meters - you can specify units)
- **Center:** Enter 0 for X, 1.5 for Y, and 0 for Z. We use 1.5m for Y because it is half of 3m, and will put the box flush with the ground at zero. It's a good idea to not only model to scale, but also to model "against the origin" so that the object touches the floor at (0,0,0).
- **Smooth Edge** - leave unchecked to conserve polygons.
- **Radius** and **Radius Segments** don't apply.
- **Segments:** Leave X and Z segments at one, but enter 3 for Y. This will give the box three "layers" from top to bottom, which we will use later to give the column shape.



You will see your box update in the viewports as you change the values. It should look like the figure shown at right – tall and skinny, with two lines across it, cutting it into thirds. When you are done, press Return to commit and create the box. Use your view controls to gaze in awe at your new creation. (Or press the “a” key to auto-fit the viewports on your box.)



### Manipulating the shape of the column

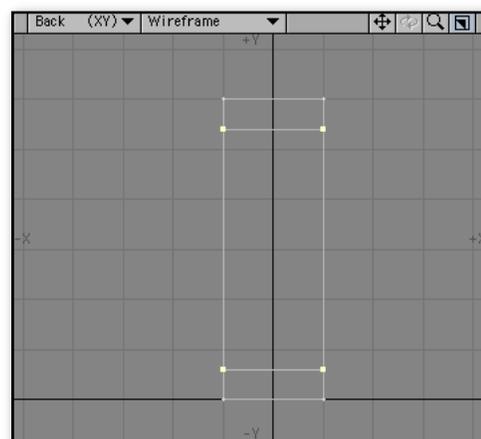
Now comes the fun part – designing the look of your column. In the left or the front view of your box, select the points that divide the top third from the middle third of your box (press the space bar until “Points” is highlighted at the bottom of the screen, then command-drag or alt-drag a lasso around that “layer” of points).

Then change to the “Move” tool. It’s in the “Modify” tool set, and can be triggered by pressing the “t” key. (Remember that one, because you’ll be using it a lot!) The cursor will change to a cross of arrows.

Now you can drag the points in any viewport to move them. Try dragging them around. Notice that modeler does some subtle snapping for you so that you can fairly easily do precise dragging, and assists you by showing you the offset of your drag in the info window in the lower lefthand window. This is useful for ensuring that you aren’t dragging the points horizontally and not vertically, but if you really want to be sure, you can control-drag the points, and it will lock the motion to whichever axis you drag on. (If you have accidentally dragged the points out of vertical, use the “undo” command in the edit menu to go back.)

Let’s drag the segment divisions up and down to make a tall middle portion and short top and bottom portions, like shown at right. Feel free to just “eyeball” this – we’re going to start relying on our sense of artistic proportion now!

Once you have dragged them to where you want, select both sets of points as shown. We now want to shrink those down so that they form a skinnier middle portion. To do this, we want to scale the X and Z directions equally (these are the two directions that form the ground).



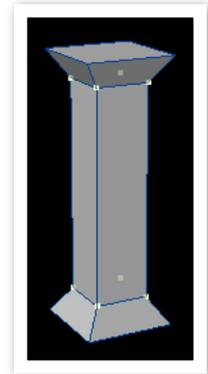
To do this, we use the “Stretch” tool (“h” hotkey) and drag on the “Top” or “Bottom” viewport. The “Stretch” tool scales the selection proportionately in the X and Z axes (or whichever axes form the viewport you are using) to how you drag. (A similar tool, “Size,” with a hotkey of “H,” scales the selection proportionately in all three dimensions.)

One tricky thing about the “Stretch” and “Size” tools is that it uses wherever you first click as the “center” for the resizing. Since we want the column to remain centered in the middle of the column, we want to resize the points around the origin.

So select the “Stretch” tool, and use the lower lefthand info window to position your mouse cursor at  $X=0$  and  $Z=0$  so that it will scale from the origin. Then mouse down and drag until the column skinniness looks good. You can keep an eye on the lower lefthand info box to ensure that the two axes are scaled equally. I chose a value of about 60%.

(Note that, as with the “Box” tool, you could instead press “n” and call up a dialog window for precisely setting the proportions and axis of stretching.)

When you’re done, you should have a column that looks like what you see at right.



### Further optimization

You may think that this is about as simple as this column could get. You would be wrong! Maybe.

It all depends on your game and what you are going to do with the object. The goal is to omit any needless geometry, so we can remove any part of the column that people will not be seeing.

Assuming this column will always be standing on the ground, we don’t need the polygon on the bottom, so let’s remove it.

Switch to polygon selection mode (press the space bar) and use the Perspective view’s rotation button to turn the column so you can see the bottom. Then click on the bottom surface. You will see it highlight in yellow.

Then press “k” to use the “remove polygons” command from the “Construct” tool set. (Or type Command-X or Alt-X, since the “cut” operation will work, too – you can cut and paste geometry, just like you cut and paste text in a word processor!) The polygon vanishes. That’s two fewer polygons Shockwave3D has to worry about when rendering your scene.

“Wait a minute,” you say. “*Two* fewer polygons?” That’s right, two. Everything in the Shockwave3D world is made up of triangles, so the square on the bottom of the column would have been translated into two triangles. Every little bit helps!

Now, depending on your game, you might also never see the tops of the columns, too, if they were always flush with a ceiling. If so, you could cut out the top polygon of the column, too. Or perhaps you are making a sidescrolling platform game, in which case, you’d also never see the back three polygons. When possible, you should optimize your objects for the task at hand.

### Saving your object

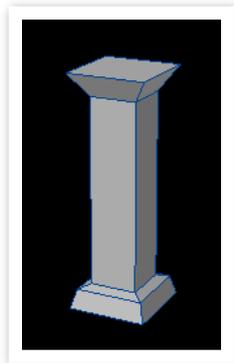
Now is a good time to save your object. Select the “Save” command from the File menu, or just hit command-S or alt-S. Save your object as “column.lwo” and put it somewhere you can find it. We’ll be using it later.

Now, suppose we weren't satisfied with our column and wanted to add some more detail. For instance, what if we wanted to have a less triangular base?

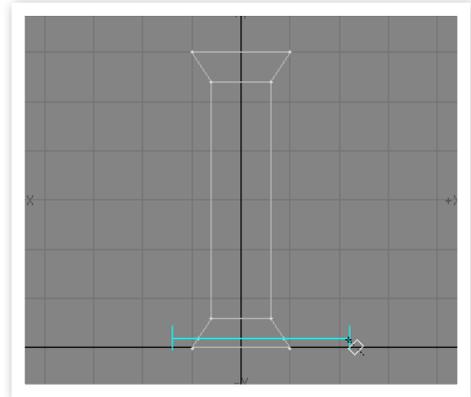
### Cutting to add detail

In order to add detail to an object, you need to add geometry – more points and surfaces which can be moved around. So let's "cut" the column near the base to add some new points that we can use. We won't actually be "cutting" the column in half, however; you wouldn't be able to drag the two halves apart. Instead, we're just adding points that weren't there before.

Select the "Knife" tool from the "Multiply" tool set (or press capital "K") and then drag a line across the base of your column as shown at right. What you are doing is defining a knife that will slice through the object along the line you define, adding geometry along the way. When you are happy with the positioning, press Return to make the cut.



You now have extra geometry that you can modify using the move tool. Switch to point selection mode and select the newly created points, and move them upward until they are almost flush with the points above them. Once you've done this, you'll have a column with a more solid-looking base, as shown at left.

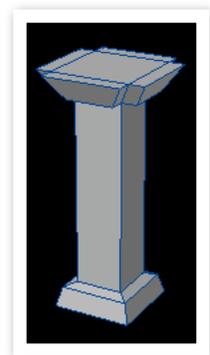


You can use this technique to add detail wherever you want. One caveat, though: you should always try to cut all the way across your model; otherwise, you will leave odd-shaped polygons that are difficult to clean up later.

### Extruding to add detail

Let's do something different with the top of the column. Let's "extrude" the sides of the column cap to give it a stronger sense of connecting with the ceiling. "Extruding" adds geometry in a different way - by pulling out one or more polygons and leaving behind surfaces to connect with the polygons they were connected to.

Enter polygon selection mode, and in your perspective view, select one of the faces of the column cap. Then choose "Extrude" from the "Multiply" tool set, or press capital "E". What you have just done is added a surface around the perimeter of your polygon. You can't see it, however, because the polygon hasn't moved, so let's move it. With the polygon still selected, select the "Move" tool ("t") and drag the polygon horizontally away from the column. You should see the column cap extend to the right, leaving behind a solid area behind it. If you watch the info window in the lower lefthand side of the window as you drag, you can remember how far you extruded the piece, and match it with the other column sides.



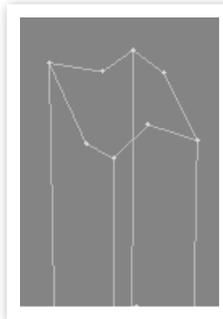
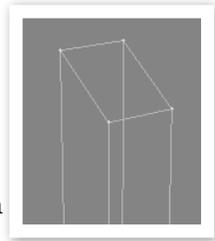
## Creating Geometry from Scratch

You might be thinking, “this is great and all, but what if I need to build something that doesn’t flow naturally from a primitive?” That’s a good question. You will often find the need to create geometry from scratch. Let’s take a look at how to do that.

Suppose I wanted to make two versions of this column: one “normal,” and one “ruined” to add some variety. Let’s do some from-scratch geometry creation to make a column that looks broken.

Save your “normal” column, and then copy and paste it into a new object file. Select “Save” from the file menu, and switch to the polygon selection mode. Make sure nothing is selected (“/”) and press control-C to copy the column. (Remember – if nothing is explicitly selected, everything is selected!) Select “new object” from the file menu, and then press control-P to paste the column into the new file. Save this new file as something like “ruined-column.lwo”. Now you’re ready to do some destruction!

Use the “Knife” tool again (“K”) to cut the column in half at an angle. Then use the polygon selection mode to select all the polygons above the cut and delete them. You now have what looks like a column that has been cleanly shaved off. But the cuts are a little too “clean” to look ruined, so let’s add some points to the edges.



In polygon selection mode, select one of the sides of the column in perspective view. Then click the “Add Points” tool in the “Multiply” tool set. This tool allows you to add points to the edges of polygons. Click once somewhere along the top edge of the polygon, and you will see (in a wireframe viewport) a point appear there. Now switch to the point selection mode, and use the move tool in the left or top view to drag the point down or up to make that edge a little jagged. Do the same for the other sides, so that each side of the column has a jagged edge. It should look something like what is shown at left.

This technique gives us points we can use to connect geometry in new ways to existing geometry. (Note: if you want to add a point along an edge shared by two polygons, be sure to select both polygons before adding points. Otherwise, the point only gets added to one polygon, and you end up with a “hole”. See the note about “well-connectedness” at the end of this article for more information.) But what if we want a point somewhere there isn’t already geometry? For instance, what if we wanted a point in the middle of those jagged edges to attach geometry to?

Let’s add a point in space. Select “Points” from the “Construct” tool set (or press “+”), and click in one of the viewports where you want the point to be located. This will define the position of the point along the two axes that the viewport defines (such as X and Y for a front or back viewport), but it only uses a default value for the depth into the viewport. You can now drag the point around in another viewport to set its third dimension position. Drag the point around in any viewport until it is in the location you want, and then press return to create the point. Use this technique to put a point approximately somewhere in the middle of the jagged edges of the column.

## Generating geometry to cover the hole

Okay, now we're ready to create geometry from scratch. As you know, polygons are simply surfaces stretched between points in space. Because of this, you can create polygons by selecting the points you want to stretch a surface between and telling Lightwave to create a polygon off of them.

### The nature of polygons

But first, we need to discuss something: polygons are *one-sided*. As you probably noticed when you cut the column, you couldn't see the polygons that were facing away from the camera. Typically, there's no reason to render polygons facing away from you because they form the surface of a solid, and the inward-facing polygons would always be covered up by the outer surface, so there's no reason to spend computing power on them.

But given a polygon with three points, how do you know which way the polygon faces? The answer depends on the "handedness" of the 3D world you are using and the order the points are taken. Hold up your hand and make a "fonzie" – a fist with your thumb pointing up. If you think about the order of points curving around from your wrist towards the tips of your fingers, the direction the polygon faces is the direction your thumb points. You can see that if you are looking at a left-hand fonzie, the polygon faces up when the points go in a clockwise motion around your thumb, and if you are looking at a right-hand fonzie, the polygon faces up when the points go in a counter-clockwise motion around your thumb.

Lightwave uses a left-handed system. So, when you create a polygon from scratch, the order of the points matters. If you want the polygon to face towards you, select the points in a clockwise fashion, and if you want the polygon to face away from you, select them in a counter-clockwise fashion.

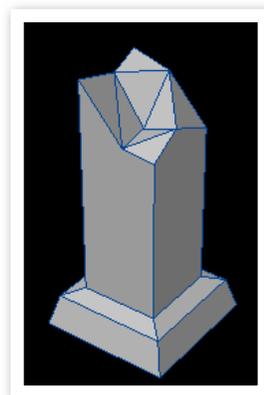
### Create some polygons from scratch

So let's try it. Select three points that you want to make a surface between. Pay attention to the order in which you select them. Then select "Make Polygon" from the "Construct" tool set (or just press "p"). A polygon should appear. If you selected the points in the right order, it will show up as expected in the perspective view when it is set to a shaded mode.

What if you get it wrong and the polygon is facing the wrong way? Well, you could delete it and do it again. But you can also just "flip" the polygon. When you create a polygon, Modeler automatically selects it for you (switch to polygon selection mode, and you'll see this). So you can just choose "Flip" from the "Detail" toolset (or just press "f") to flip the polygon around.

(As a side note, did you notice when you select polygons in wireframe mode that there are little yellow lines sticking out of them? That's the "surface normal," which tells you which direction a surface is facing. The "surface normal" points in the direction a surface is facing, so you can always check the direction of a polygon, even when it is not shaded.)

Continue selecting points and creating triangles until you have sealed off the top of the column. You're done! It should look like what's shown at right.



## Wrap-up

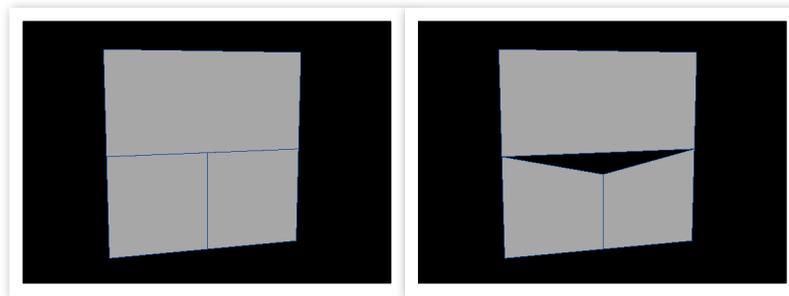
You now have all the skills you need to create basic low-polygon models for games. Using a combination of the previous techniques, you can create most scenery objects you might need for a Shockwave3D game.

However, if these were the only tools you used, you wouldn't be very efficient. This is just a starting point. Practice with these techniques, but explore some of the other tools that Modeler offers, because the more tools you understand, the faster you will be able to create models you want.

## Specific tips for modeling objects for Shockwave3D

Finally, here are some tips for modeling objects for Shockwave3D content. Some of them may not make sense to you right now, but as you gain experience with it, these tips may come in handy.

- Director requires that all polygons be triangles. The exporter will break polygons with more than three corners into triangles for you, so you do not need to work solely in triangles, but polygons that are nonplanar with more than three points can “break” in different directions, so use your Info tool to find nonplanar polygons and manually split them so that there are no ambiguous cases for the exporter. Similarly, non-convex shapes are better broken up explicitly before export.
- Models should be “well-connected”. You will notice “tearing” in your models if they do not form continuous surfaces. Even if a surface *appears* continuous, it may not be; if you can drag a point to create a hole in a surface, as shown below, then the surface is not well connected.



*This plane will show “tearing” once brought into Director, because you can drag a point to leave a hole.*

- When a model gets moved into Director, it may seem very large. This is because one meter in Lightwave translates into 100 world units in Director (with the default settings). Unfortunately, Lightwave does not allow for a default unit of centimeters, so the easiest approach is to simply model normally in Lightwave, and just think of the world units in Director as centimeters.

## More Information

For more information on developing Shockwave3D-based games using Lightwave and Director, visit the Shockwave3D Developer's Guide at <<http://homepage.mac.com/nephilim/sw3ddev/>>.