Introduction to Building in OpenSimulator

Edited by Chris M. Collins (Avatar: Fleep Tuque)

This curriculum is adapted from **Global Kids' Second Life curriculum**, previously hosted on RezEd.org, with thanks to **Joyce Bettencourt** and **Barry Joseph** who started the project. Modifications and adaptions were made in the **"GKCx"** update **by Jeremy Kemp and Stephen Kemp,** released in 2011 at simteach.com/gkcx under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Both sites are no longer available on the web. Some sections were also modified or written in whole or in part by the **University of Cincinnati Center for Simulations & Virtual Environments Research (UCSIM)**. The introduction to scripting chapter was written by **Jeffrey Steele (Ryon Bloobury)**.

Many thanks to all who have contributed text, images, or open source items over the years to make this great resource for new builders! =)

For more information about Fleep's OpenSimulator projects, visit http://fleeptuque.com and http://fleepgrid.com.

What is OpenSimulator?

OpenSimulator is software that simulates a virtual world very similar to our own physical world.

In the picture, the OpenSimulator software is what makes the island beneath our feet solid, the water waves twinkle in the simulated sun, and the clouds drift slowly overhead. A gentle breeze blows the virtual air, making flags and cloth items flutter, and an object placed in the air will fall to the ground, just like gravity does in the real world. OpenSimulator is the collection of computer programs needed to simulate all of the simple and complex interactions that can take place in this virtual environment.

OpenSimulator is *open source* software, which means anyone is free to download and install it on their own computer or server, and anyone can look at the computer program code and edit or improve it if they want to. In fact, the OpenSimulator software is written and created by many people all over the world who work together to make the software better, and anyone who wants to help improve the software is welcome to join the community of OpenSimulator developers.

For the technically-savvy, this is the description of OpenSimulator from theofficial wiki:

OpenSimulator is an open source multi-platform, multi-user 3D application server. It can be used to create a virtual environment (or world) which can be accessed through a variety of clients, on multiple protocols. It also has an optional facility (the Hypergrid) to allow users to visit other OpenSimulator installations across the web from an account on a 'home' OpenSimulator installation.

OpenSimulator allows virtual world developers to customize their worlds using the technologies they feel work best - we've designed the framework to be easily extensible. OpenSimulator is written in C#, running both on Windows over the .NET Framework and on Unix-like machines over the Mono framework. The source code is released under a BSD License, a commercially friendly license to embed OpenSimulator in products. If you

want to know about our development history, see History.

Out of the box, OpenSimulator can be used to simulate virtual environments similar to Second Life™, given that it supports the core of SL's messaging protocol. As such, these virtual worlds can be accessed with the regular SL viewers. However, OpenSimulator does not aim to become a clone of the Second Life server platform. Rather, the project pursues innovative feature development with an aspiration towards becoming the bare bones, but extensible, server of the 3D Web.

OpenSimulator is getting more stable as it approaches release 1.0, but we still has a few quirks; handle with care!

Features

- Supports online, multi-user 3D environments as small as 1 simulator or as large as thousands of simulators.
- Supports 3D virtual spaces of variable size within one single instance.
- Supports multiple clients and protocols - access the same world at the same time via multiple protocols.
- Supports realtime Physics Simulation, with multiple engine options including ODE.
- Supports clients that create 3D content in real time.
- Supports inworld scripting using a number of different languages, including LSL/OSSL, C# and VB.NET
- Provides unlimited ability to customize virtual world applications through the use of scene plugin modules.

For the purposes of this manual, you don't really need to know all those technical details, it should be enough to know that OpenSimulator is the software that makes the water, land, and sky in this virtual world - and all the things you can do in it - possible!

Getting Started

Welcome to Building in OpenSimulator!

Of all the wonderful things that virtual worlds offer, the ability to create, prototype, play, and build whatever we can imagine is by far the most compelling feature.

Before the invention of platforms like OpenSimulator, 3D modeling programs often required professional level training to create 3D models, and even after you mastered the complex techniques required to build virtual objects, the objects lived inside the graphics program used to create them. You couldn't invite a friend to come walk around your creation with you, or fly into it and have a conversation about how it works.

OpenSimulator allows even a novice to learn how to create virtual objects, and since the platform is multi-player, others can see your creations in real-time, even build collaboratively with you! This ability to turn the visions of our dreams and imaginations into virtual reality, and share it with others, is what makes OpenSimulator so magical!

What should I already know before I start learning to build?

This manual does not cover certain basic skills, such has how to navigate, move your camera, or operate your viewer interface. We are assuming that you already have some basic level of skill in how to move around and look at the virtual world.

Learning how to navigate (walk and fly) and operate your camera with skill and precision are necessary prerequisites to learning how to build.

If you don't already have these basic skills, you may wish to practice or search for tutorials on the web before proceeding with this manual.

Ok, I know how to walk around and move my camera. Where can I start building?

Before you can begin to build, you must be logged into an OpenSimulator grid, and be on land that you either own, or have been given permissions to build upon.

Note that just because you can build somewhere doesn't mean that you should!

Most grids will have a Terms of Service, Community Standards, or other rules and policies that govern who can build where. Make sure you familiarize yourself with the rules of any grid you log into so you understand what is and is not permitted. Building on land that you don't own, or don't have permission to build on, may be interpreted as a form of *griefing*, and may result in disciplinary actions or even getting banned from the grid.

If you aren't sure where to build, look for a **Sandbox,** which is typically an area where grids permit the public to build.

It is beyond the scope of this manual to explain how to log into an OpenSimulator grid, but generally, you need the following:

- Download, install, and run a viewer compatible with OpenSimulator
- If logging in locally to a grid, you will need to create an account and then configure your viewer with the Login URI of the destination grid
- If hypergriding, you will need an account on your home grid, then enter the Hypergrid address of the destination grid into your viewer map search

For more information about connecting to a grid, see the OpenSimulator wiki viewer page.

What viewer should I use to follow the tutorials in this manual?

This manual uses the Singularity Viewer for pictures and images.

At the time of this writing, the images in this manual were created using the Singularity viewer. If you are new to OpenSimulator and don't already have a viewer preference, you may wish to download and install the Singularity viewer so it is easier to follow along with the examples.

If you prefer a different viewer, you should still be able to follow the tutorials, but certain menu items or settings may be in a slightly different place or use a different name. For assistance, look at the help pages for the viewer you are using.

Ok, are you ready to get started turning your dreams and visions into virtual reality? Turn the page!

Basic Building

Building in OpenSimulator takes skill and practice, but every tree, building, piece of furniture, vehicle, and any other object begins with a simple "prim" or primitive shape. You might consider prims the "legos" of OpenSimulator.

This chapter will walk you through the skills you need to create your own objects in-world, from basic prim manipulation all the way to complex techniques. A few things to keep in mind as you get started:

- Always make sure you are building on land with the landowners permission. Just because you technically **can** build doesn't mean you **should.**

- It is critically important that you always **NAME** your items, give the item proper **PERMISSIONS**, and take a copy to your inventory to **SAVE** your builds. You will learn how to perform these tasks in the tutorial, but just remember that failing to name, permission, and save your work can mean hours and hours of lost effort.

- One of the best ways to learn how to become a great builder is by inspecting the builds of others. You can right click > Edit any object in the world to see how it was constructed!

Topics in this chapter include:

- Introduction to Building
  - Moving and Rotating a Prim
  - Resizing a Prim
  - Linking a Prim
- Building: Playing with Shapes
  - Using Path Cut to Slice Prims
  - Hollow and Hollow Shape
  - Taper, Top Shear, and Dimple
  - Twisting Prims
- Building: Additional Features (Parameters)
  - Flexi Prims
  - Adding Light

Introduction to Building

Ok! Time to start learning how to build with the basics of creating and manipulating basic prims!

By the end of this module, you will have the following skills:

- Creating (rezzing) a basic building block (a prim)
- Moving and rotating prims
- Changing the size of a prim
- Joining (linking) several prims together to form a more complex shape
- Cloning prims

You will demonstrate your skills by:

- Rezzing prims of different shapes
- Moving and rotating prims
- Changing the size of a prim
- Making an object that uses several prims
- Duplicating shapes and objects that you make

BUILDING: INTRODUCTION

One of the most enjoyable things to do in OpenSimulator is to build your own environment. The interface building tools are available for everyone to use. Your creations can be saved, shared, modified and even bought and sold. Learning how to build can seem intimidating at first. Not to fear! Beginning with this chapter, you will learn - step by step.

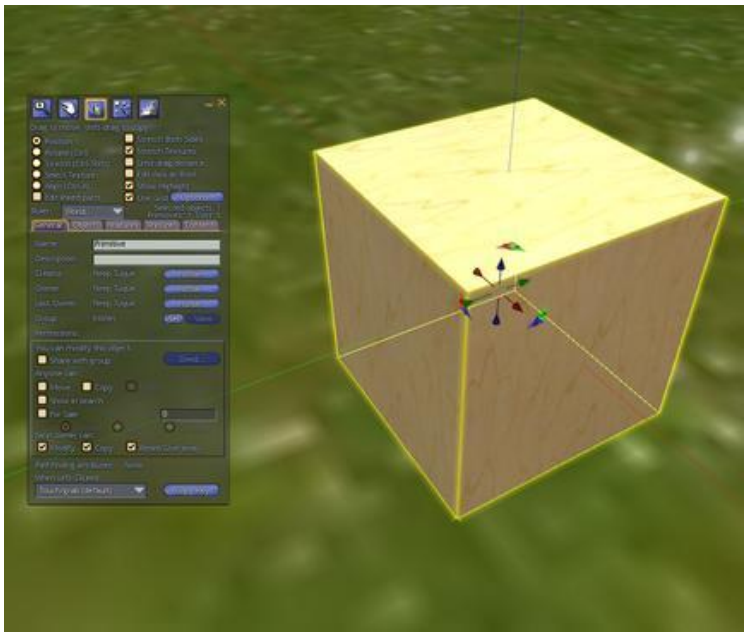
Every object you see in OpenSimulator - from the common cube to the most amazing vehicle or building - has been built, or created, using tools that are available to you. In this mission, you will learn how to find a place to build and create basic shapes, called primitives, or prims for short.

A Note about WHERE you can Build: Remember, you may build only in certain areas: 1. On land you own 2. On land owned by a group you belong to 3. On land owned by a person or group that has given you rights to build on their land 4. In Sandboxes - land set aside just for building.

Practice Building Skills

Step 1: Rezzing a prim
The basic building block in OpenSimulator is called a *prim*, which is short for primitive. Creating, or making something appear, is known as *rezzing*.

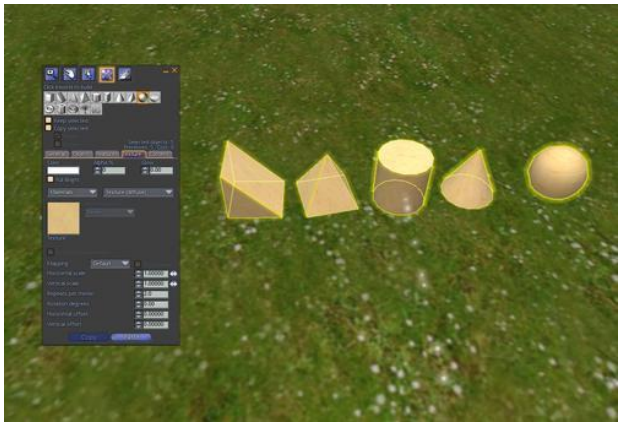
To rez a prim, open the Build menu (B or CTRL+B depending upon your viewer). You will see your cursor turn into a wand. Click on the ground where you would like your prim to appear, or rez. By default, you will see a plywood cube appear.



Try this now: Rez a cube!

Step 2: Rezzing other shapes using the Edit window

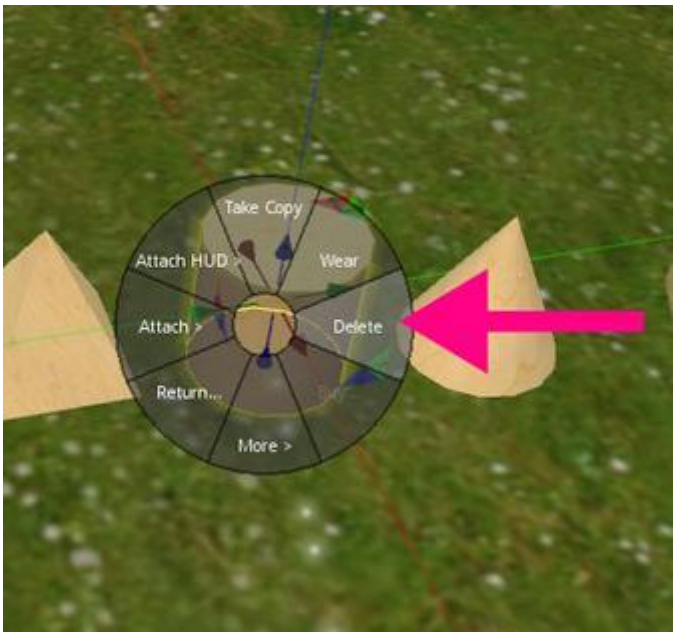
You are not limited to cubes. In fact, you can choose to rez several shapes to use in your building. When you created your first cube, you may have noticed a new window open on your screen. This is the Edit window. To rez a different shape, open the Build menu again (B or CTRL+B depending upon your viewer) and when the Edit window appears, select one of the other shapes you see. Then click on the ground with the wand.



Try this now: Practice rezzing prims using the different shapes available to you.

Step 3: Cleaning up after yourself

It is important to leave room for others to build. When you are through rezzing prims, right-click (Cmd-click) on each of your prims and choose Edit, and then Delete from the menu (may need to go to More > Delete depending upon your viewer), or delete using the Delete key on your keyboard.



It's considered courteous to always clean up after yourself, so make sure to clean up your objects before you leave!

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Moving and Rotating a Prim

Congratulations! You can make building blocks! However, just as with regular building blocks, you need to learn how to move them around to create complex objects.

Instruction

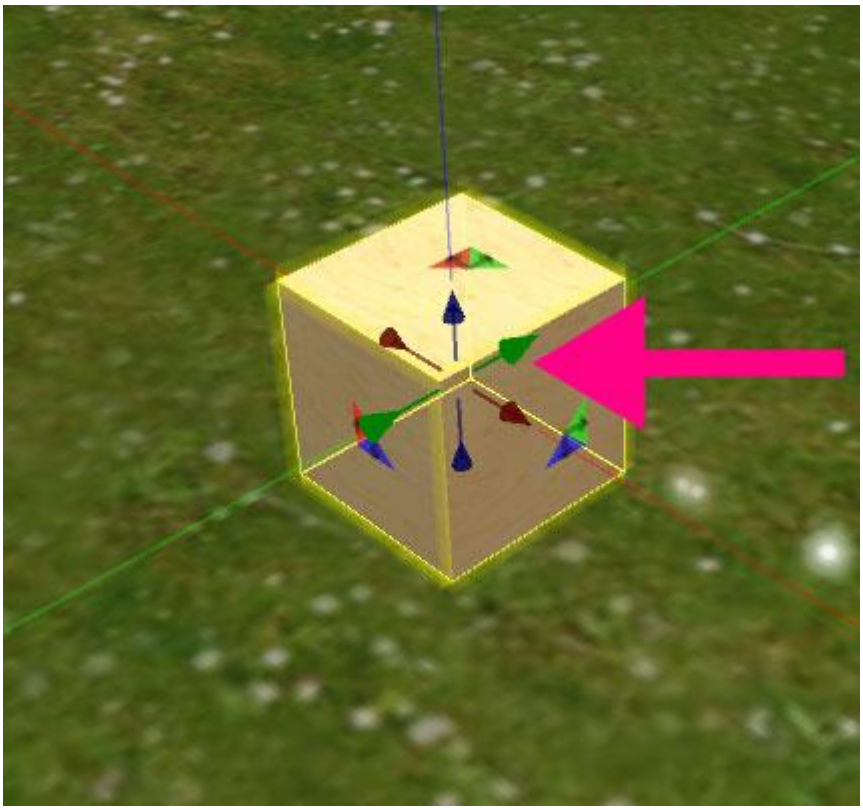
OpenSimulator offers two primary ways to move and rotate objects. You can drag an object around and just "eyeball" it where you want it to go. Or, you can use numbers to place it exactly where you want it to go. Although you may find you prefer one way more than the other, a good builder knows them both.

Practice

Step 1: Dragging a prim
Rez a cube near you. You will see red, green and blue directional arrows showing the x, y and z axes.

Red: x axis
Green: y axis
Blue: z axis

If you place your pointer over the head or tail of an arrow, you will see it glow more brightly.



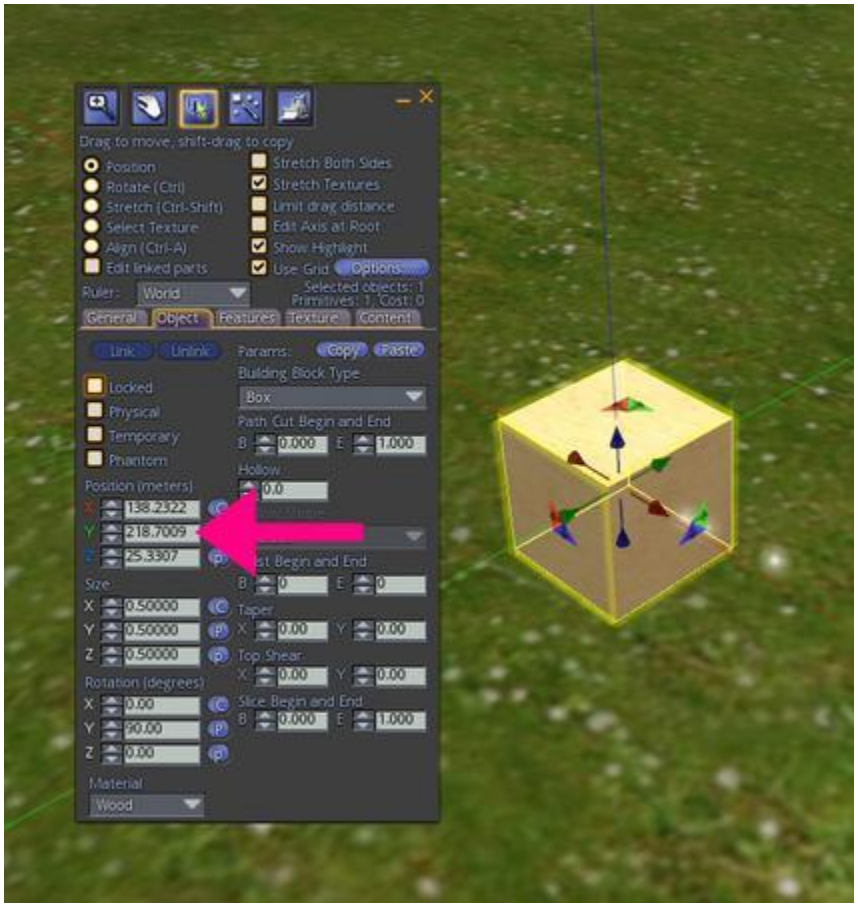
Click and hold to drag your cube along that axis. You might have to try a few times to get it right, so be patient and point with precision. Hint: you can move your camera angle if necessary to get a better shot.

Practice dragging your prim using each of the directional arrows. Also try dragging using a two-colored arrow if one is visible; this will allow you to move a prim diagonally.

Step 2: Using numbers to position a prim

A prim's position can be described mathematically, using a number for each axis. The number represents exactly where a prim is located on a section of land, called a sim, which stands for simulator, also sometimes called a "region" or an "island" in OpenSimulator.

To see the position of a prim **Right-click** on an existing prim, choose **Edit** and select the **Object tab** in the Edit window. (May appear as the Options tab depending upon your viewer.)



When you want to line objects up exactly, building by the numbers is very helpful. You can see these numbers using the Edit window. To do this, either create a new prim, or Right-click (Cmd-click) on an existing prim you've created and choose Edit. Select the Object tab to view the mathematical position of your prim. Edit Window > Object Tab > Position To change an object's position, you can type in a number or use the up/down arrows to the left of each number.

Use the up/down arrows to the left of each number to slightly adjust the position of your object. Notice how far an object moves using only small adjustments. Depending on the axis you choose, your prim will move the following way:

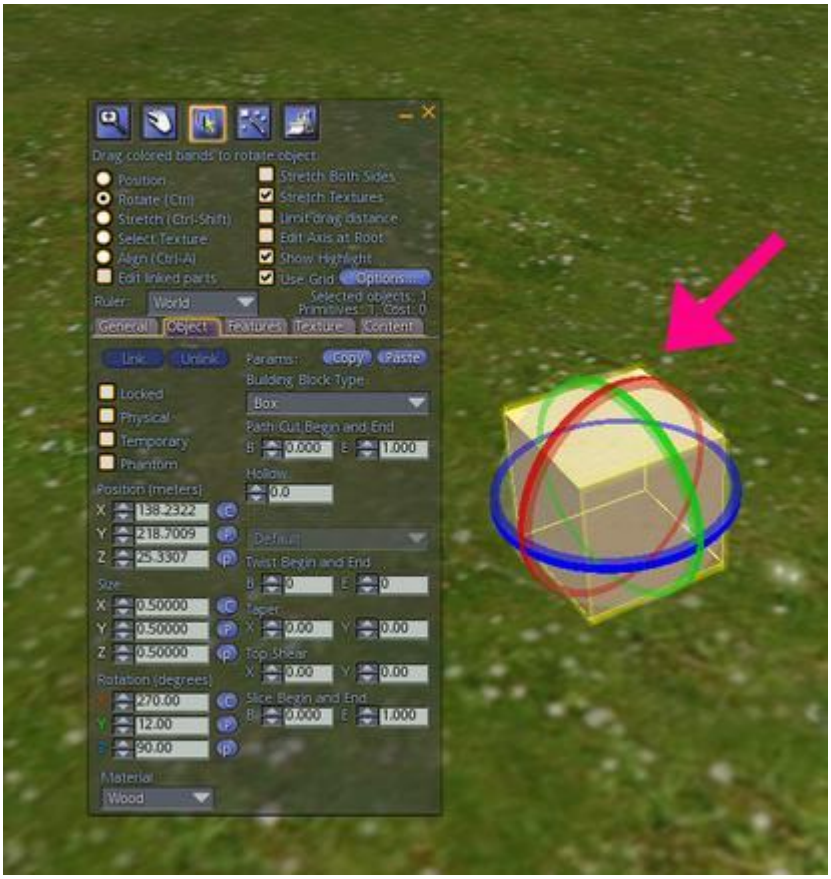
X (Red): East or West
Y (Green): North or South
Z (Blue): Up or Down

Tip: If you type in a new number, make only small changes at first. If you drastically change an object's position, you may not be able to find it easily!

Step 3: Rotating a prim

To rotate a prim, make sure you have it in Edit mode. If you can't see the directional arrows, Right-click (Cmd-click) on it and choose Edit from the menu. Then hold down the **CTRL** key and you will see the arrows change to colored circles. Dragging on each of these circles allows you to rotate a prim. You may also select the Rotate button from the Edit window to show the colored circles.

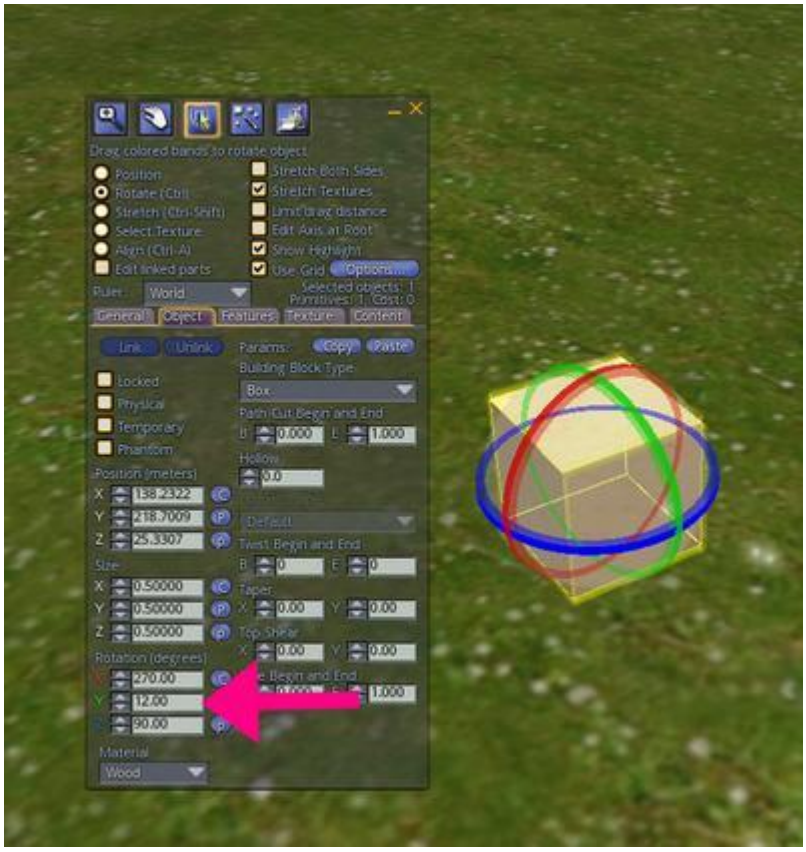
From the edit mode, hold down the **CTRL** key and click and drag the circles around your prim and rotate it to your liking.



Step 4: Using numbers to rotate a prim

A prim's rotation can also be described mathematically. If you remember common angle degrees from geometry, you will see how they look in OpenSimulator! Select the Object Tab to view the rotation degrees of your prim. Edit Window > Object Tab > Rotation To change an object's rotation, type in a value between 0 and 360 degrees. You can also use the up/down arrows to the left of each number to change the rotation one degree at a time.

Change the numbers of each rotation axis and notice what happens to your prim.



Take some time to move and rotate all the prim shapes. Just using the shapes you know, can you make an ice cream cone? What else can you make?

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Resizing a Prim

Now that you've seen all the different basic shapes you can rez, it's time to learn to change their size.

Instruction

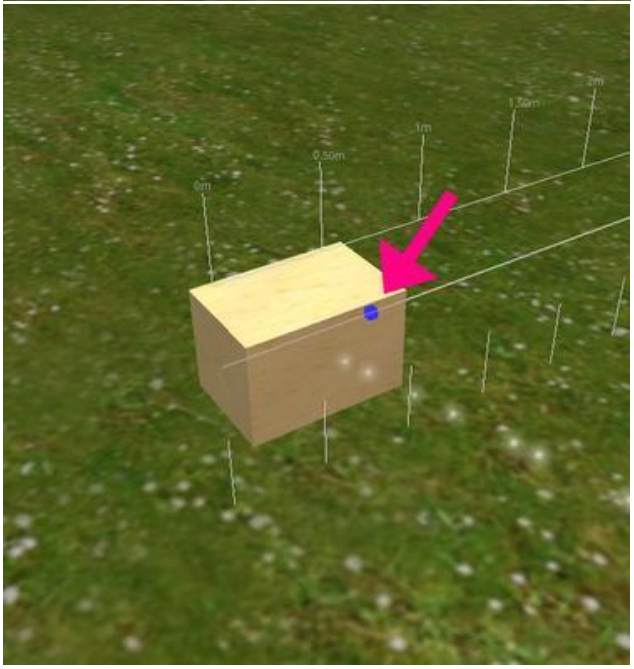
You can resize a prim to be larger or smaller. You can also change a prim by changing just one dimension of it. For example, you can begin with a cube, and s-t-r-e-t-c-h it into a rectangle. Just like moving and rotating a prim, you can eyeball a prim's shape and size, or resize it mathematically. This lesson will teach you how to do both.

Practice

Rez a cube prim on the ground and hold down the **CTRL+SHIFT** buttons. You should see the arrows on your cube disappear and small colored boxes will appear on each face of the prim, and small white boxes will appear at the vertices of your prim.

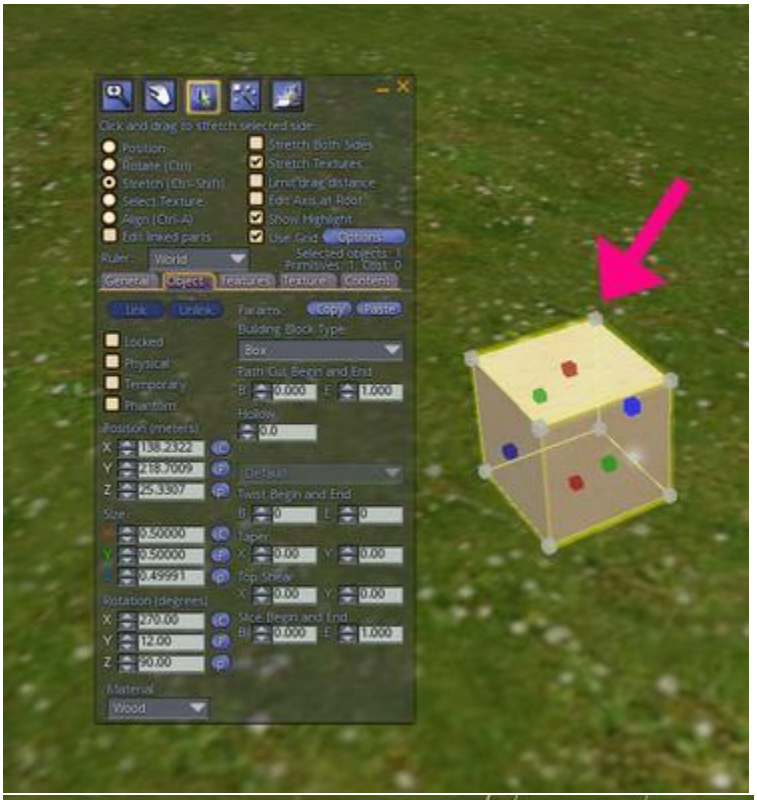
Step 1: Resize a Prim Along One Dimension

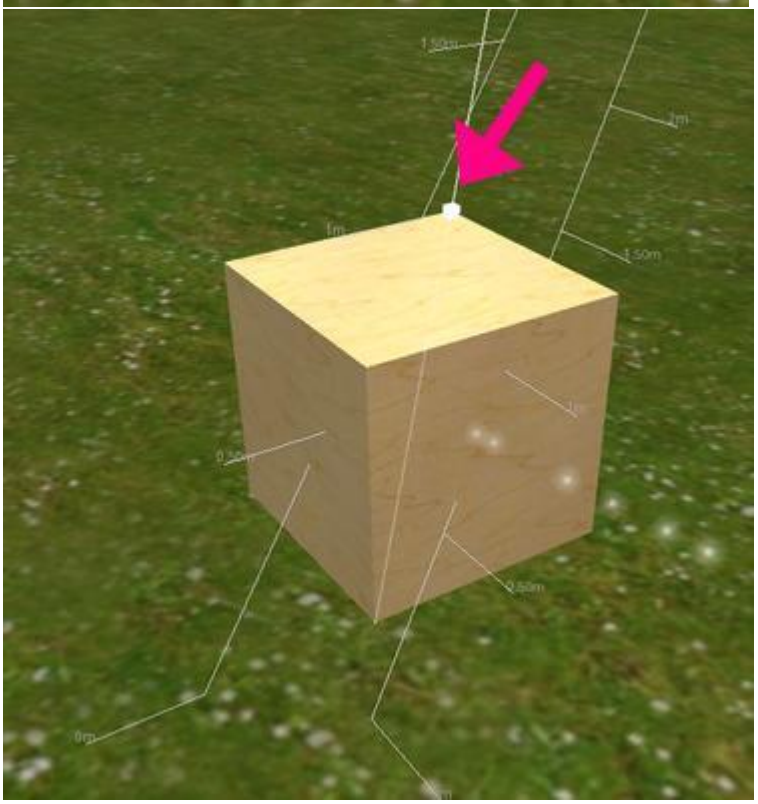
To stretch the prim along a particular dimension, still holding down the **CTRL+SHIFT** keys, left click on one of the colored boxes and click and drag to stretch the prim.



Step 2: Resize a Prim Proportionally

To resize the prim in proportion, click and drag on one of the white boxes at one of the vertices (corners) of your prim.



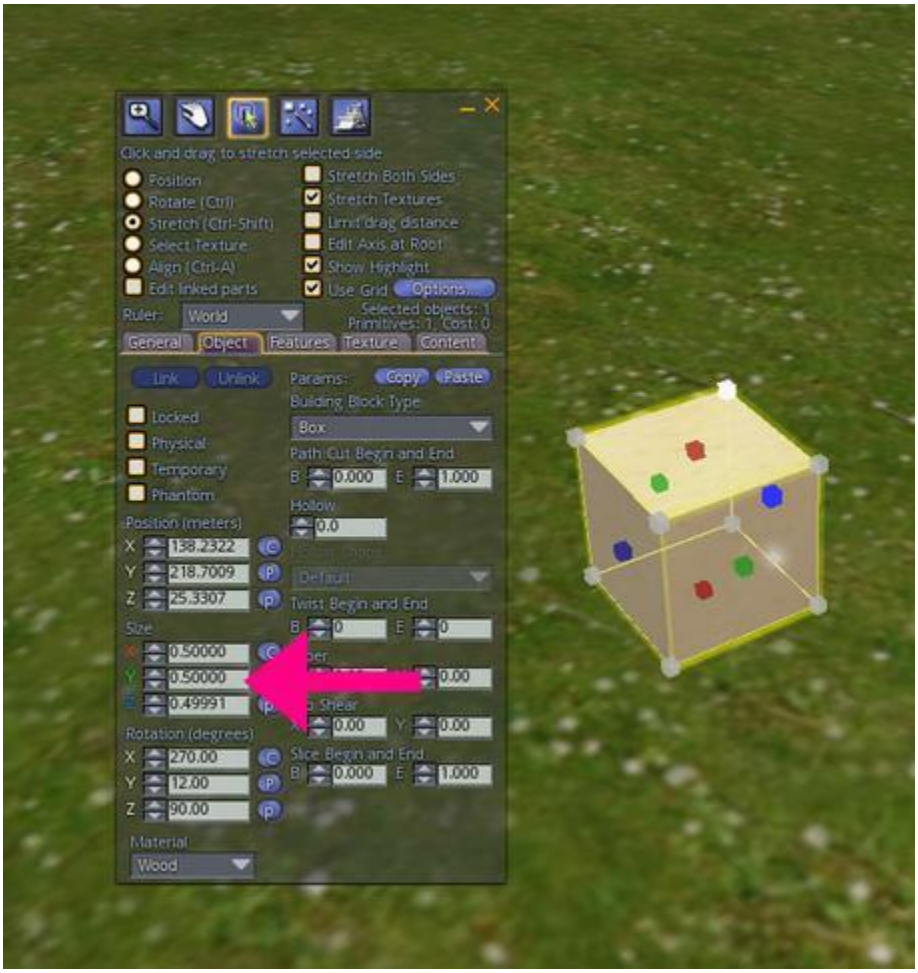


Try this now: Practice dragging the different colored squares to alter both the shape and size of your cube. Try making a large flat rectangle, tall thin tower and everything in between.

Step 3: Using numbers to resize a prim

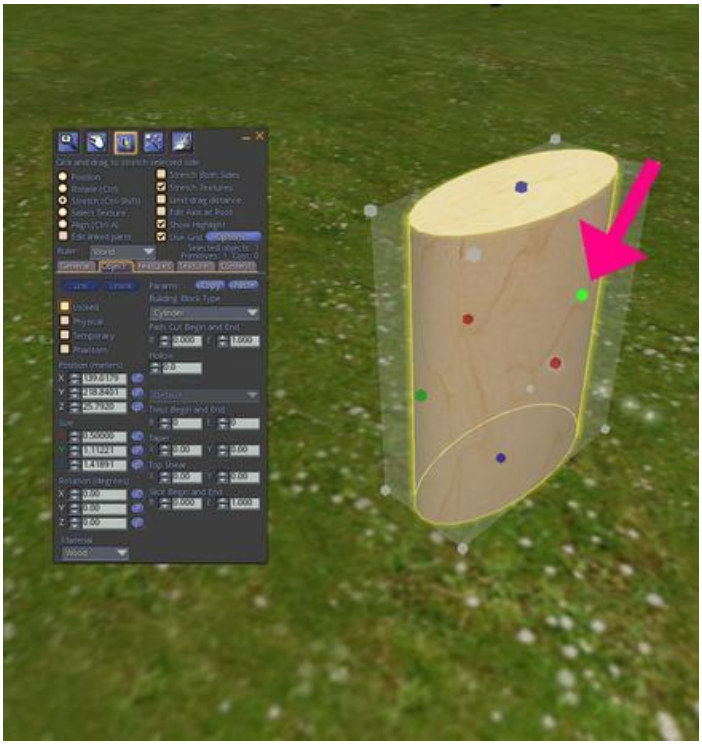
Just as with positioning and rotating a prim, you can use exact numbers to size a prim. A prim is always rezzed a particular size. A cube, for example, always begins as half a meter (0.500) on each side.

Practice entering different values for each of the axes to see how they affect the size and shape of the cube.



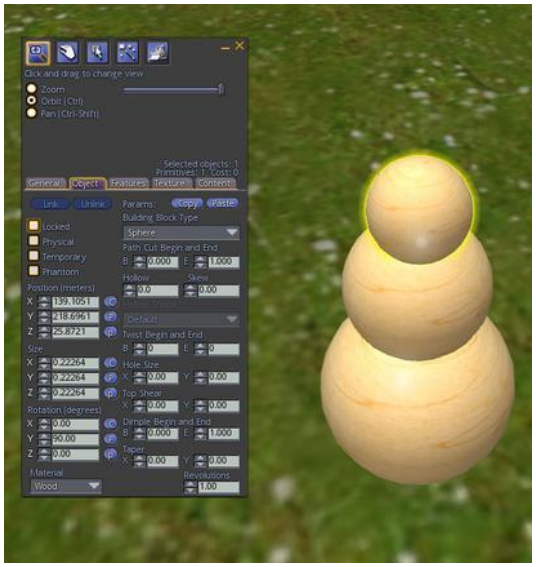
Step 4: Practice stretching and resizing other shapes

Other shapes stretch and size in slightly different ways. ***Try this now:*** Take a few minutes to see how you can alter the other shapes you have to work with.



Step 5: Build a Snowman

Now you can stretch and make shapes different sizes. Can you put them together to make a more complex shape? A snowman, for example?



Take some time to create something that takes several prims. You can move prims so they are partially inside each other to create really interesting shapes! When you are done experimenting, make sure to take a few minutes to clean up!

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Linking a Prim

In the last section, you started to put together a more complex object using at least a few prims. You may have wished to "glue" it together somehow and keep it. But how? That is what you will tackle in this section!

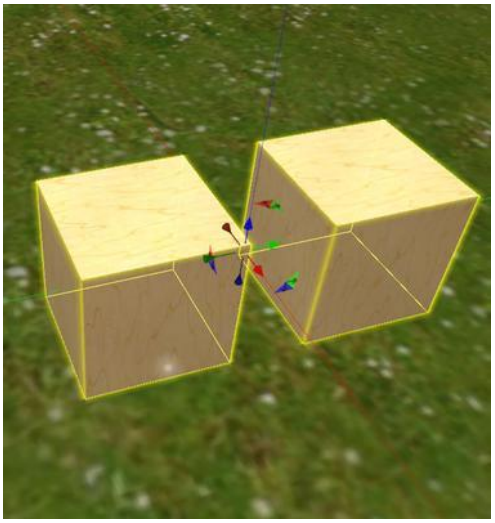
Instruction

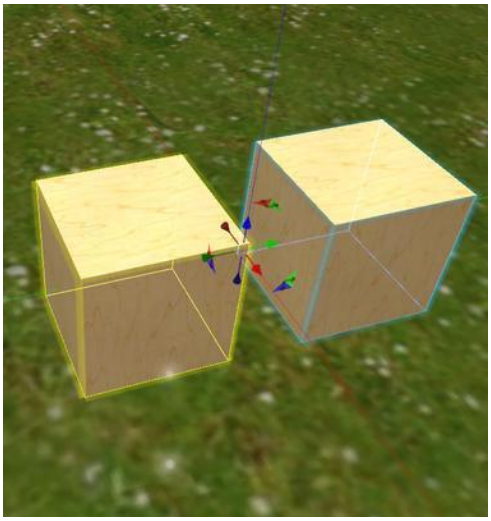
Linking is the process of attaching prims together so they can be moved as one object. In this section you will learn how to link and unlink prims. You will also learn how to edit a prim, even when it's linked to another prim. Finally, you will learn about root prims and why they are important.

Practice

Step 1: Linking two prims

Rez two prims close to each other. Right-click (Cmd-click) on one of the prims to select it and choose Edit mode. Holding down the **Shift** Key, click on the other prim. You will notice that they are now both highlighted in yellow. Now hit **Ctrl-L** to link them both and you will see that moving one prim moves them both together, and one prim is highlighted in yellow, the other in blue.



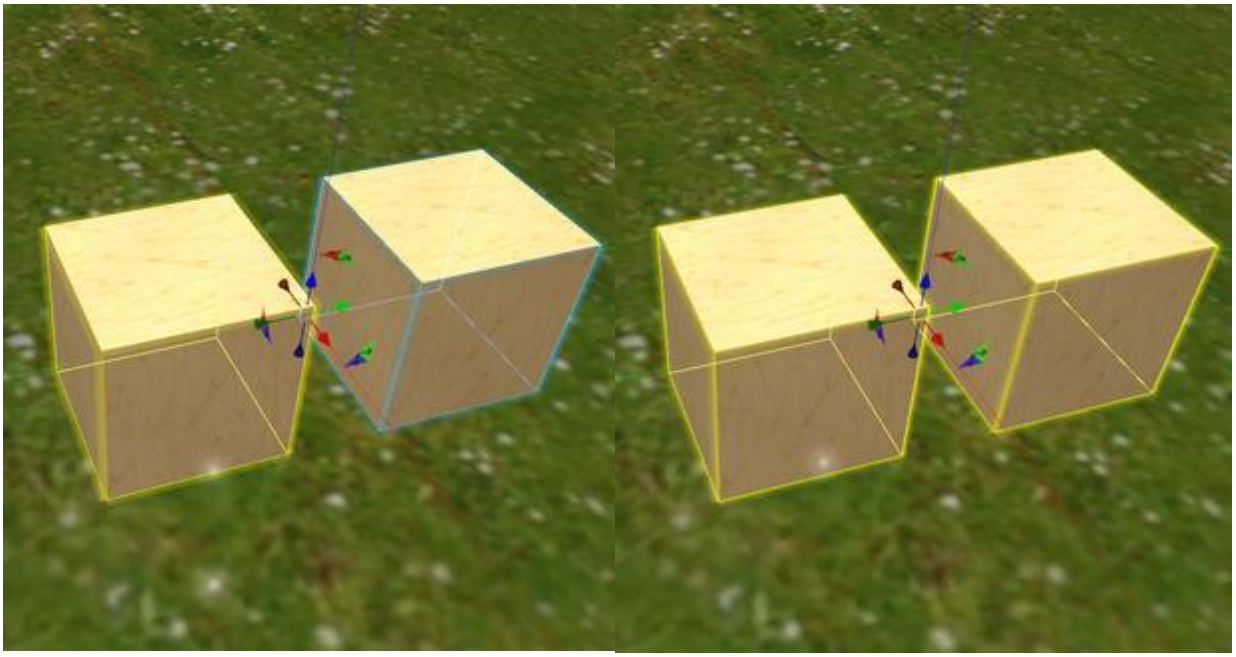


Move and rotate them around. Note: If prims are too far apart, you will get a message saying they cannot

be linked. The smaller the prim, the closer it must be to another prim to be linked.

Step 2: Unlinking a group of prims

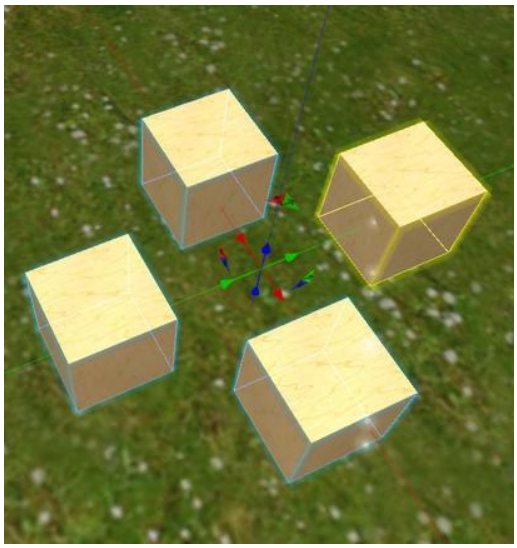
To unlink a group of prims, Right-click on them and hit **Ctrl-Shift-L.**



Unlink the two prims you just linked.

Step 3: Linking more than two prims

If you are working with many prims, you can Right-click (Cmd-click) on the first one, press the shift key down and carefully shift-click on each other prim. When you are certain you have selected all of them, hit Ctrl L to link all the prims together.



Rez at least three or four prims and link them all. Check to see if they are linked by moving them as a group. If one gets "left behind", you can select the group of prims and then shift+click on the ungrouped prim and link it as well.

Step 4: Editing just one prim

To change or delete just one prim in a group, select Edit Linked Parts from the Edit window. Then, click on the prim you want to edit.

Use the Edit Linked Parts button to edit just one prim. Try adjusting the position, resizing or rotating it.

Step 5: Unlinking just one prim

To unlink a single prim, make sure Edit Linked Parts is selected in the Edit Window. Click on the prim you want to unlink. Then hit **Ctrl-shift-L** to unlink. Now, you can delete that prim without it affecting the other linked prims.

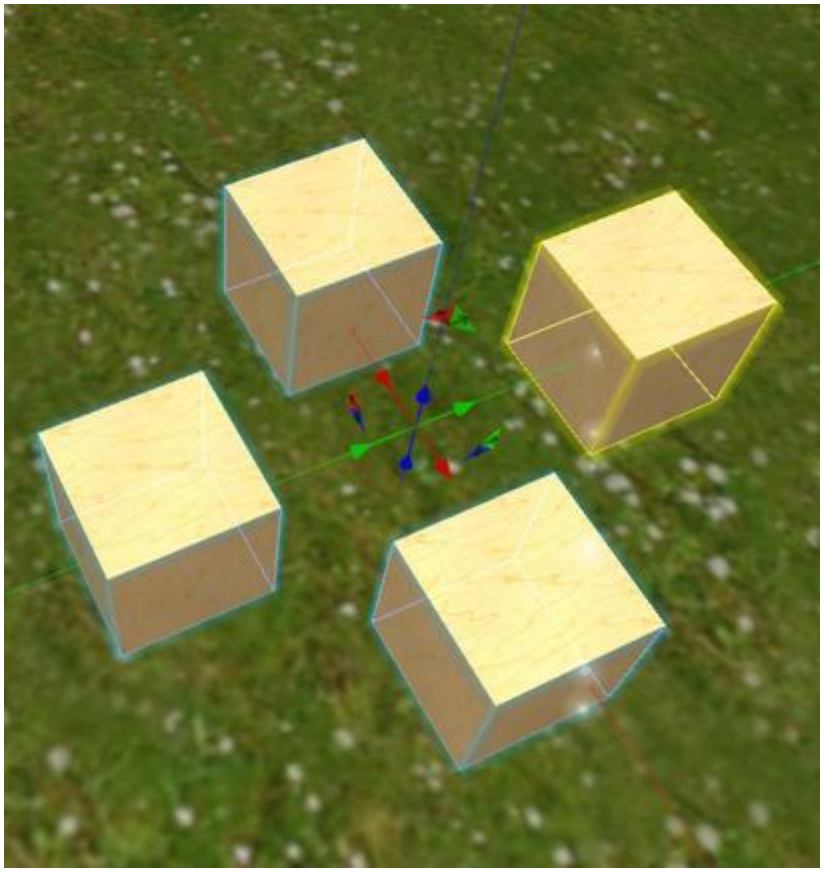
Select, unlink and delete just one prim in a linked group. Make sure you can do this without unlinking all the other prims!

Step 6: About Root Prims

A root prim is the last prim you select when linking a group of prims and has some special qualities you will want to be aware of:

- The position of a group of linked prims is based on the location of the root prim.
- The name of a group of prims is based on the name of the root prim.
- Some scripts will act on the root prim in a special way.

To find the root prim of a linked object, Right-click (Cmd-click) on it so that it is in Edit mode. All the prims will be highlighted in blue except the root prim, which will be highlighted in yellow.

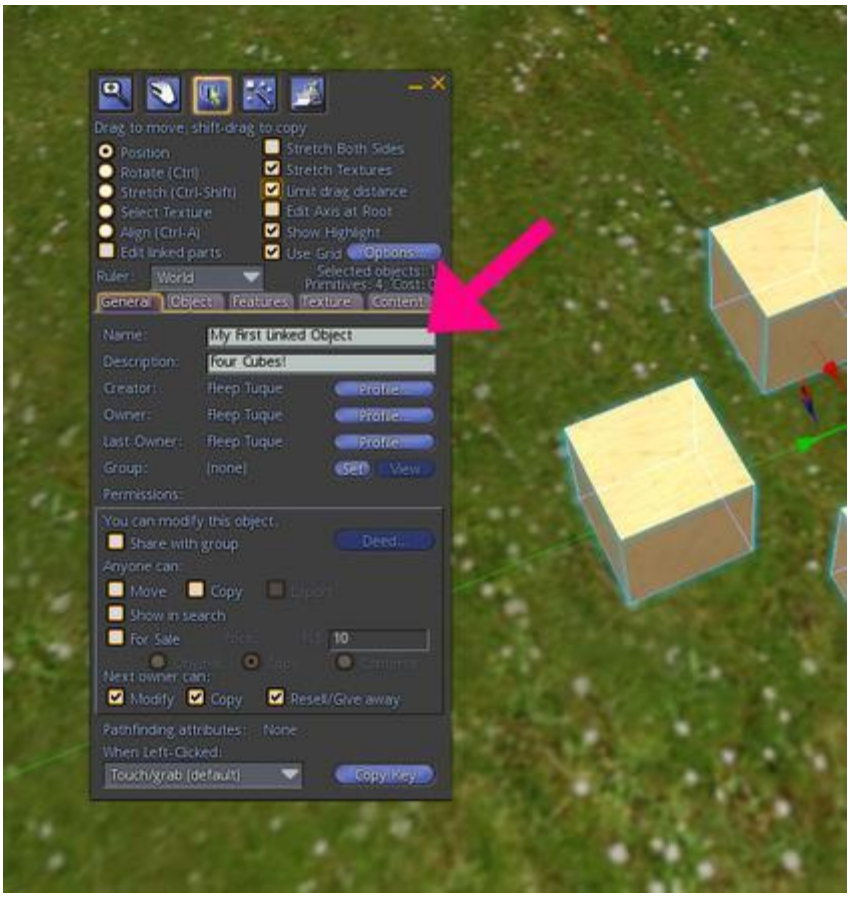


Find something that looks complicated enough to be made from more than one prim. Right-click (Cmd-click) on it and choose Edit. You should see each prim highlighted, with the root prim in yellow. Note: This is a wonderful way to see how objects have been made!

Step 7: Renaming and keeping your new Object

By default, any group of linked prims is called Object. As soon as you create and link something you think you'd like to keep, you should rename it with a descriptive name.

To rename an object, Right-click (Cmd-click) on it and choose the General tab from the Edit window. Double-click on Object in the Name field and type in your new name. You may also type a description in the Description field if you wish. Now, when you put your new creation into your Inventory, you'll be able to find it much more easily.



Rez three prims, and name each prim "Prim 1" "Prim 2" and "Prim 3" in the Edit > General > Name field. Then link the three prims together noting which prim you selected as the root prim and what name the linked object has when you are finished. Then Right-Click > Take to take the item into your inventory.

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

That's the end of the first section of this building tutorial!

Playing with Shapes

Cubes, spheres and pyramids are all great shapes to use for building. By now, you've probably looked at amazing and unusual shapes and wondered, "How did they DO that?" In this section, you will learn how to edit these basic shapes to make fantastic new building blocks to use in your builds.

TERMS*:*
Taper: Decreasing the size of one end of an object.
Shear: Cutting the end off an object at an angle.

Learning Objectives:
By the end of this module, you will have the following skills:

- Rezzing different shapes in different ways
- Cutting and hollowing prims
- Tapering and shearing prims
- Twisting prims

You will demonstrate your new skills by creating amazing new shapes to use in your own creative builds.

Different Ways to Rez Basic Shapes

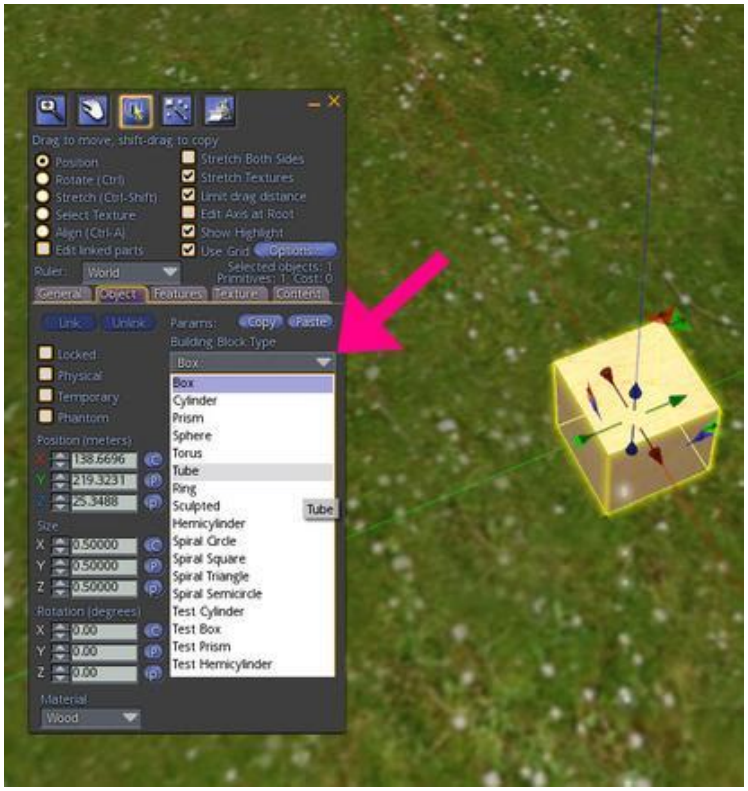
All building starts with putting together basic shapes. Just like a real set of wooden building blocks, or legos, different shapes can be stacked and put together to form new shapes. However, in OpenSimulator, you can start with a basic shape and change it into something very different.

Instruction

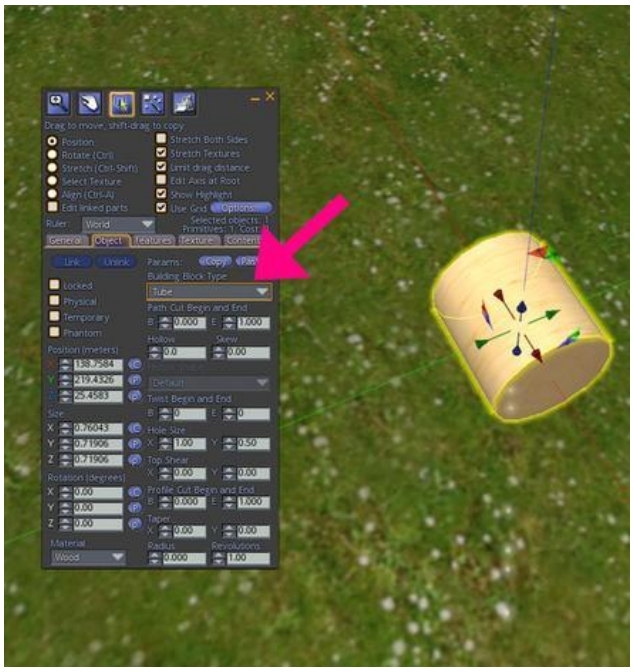
There are two main ways you can rez a basic shape. You can use the Edit window to create a basic shape or you can create a default cube and then change its shape.

Practice

Step 1: Changing a cube to another basic shape
Begin by rezzing the default cube and then changing it to another basic shape. With a cube in Edit mode, open the Edit window to the Object tab.



In the Edit window, on the left side of the Object tab, is a dropdown menu: Building Block Type. Click on the down arrow to see a menu of basic shapes. Clicking on one of those shapes will change the cube to that shape!



Right-click (Cmd-click) on the ground, choose Create from the pie menu and rez a cube. With the Edit window open to the Object tab, choose the Building Block Type drop-down menu and choose another shape. Now click through the other shapes to see what happens. If you try to go back to your original shape, you may find it has changed. Try clicking back on the Box shape to see what it looks like now. Note: A Sculpted prim shape is a special kind of prim that will be covered in a later building module. For now, just explore the basic shapes.

Make sure you take the time to play around with all the shapes. Spending some time now will make you a better and more creative builder in the long run.

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Using "Path Cut" To Slice Prims

Building is just going to get cooler and cooler. What if you could just cut a "chunk" or a "wedge" out of a shape?

Instruction

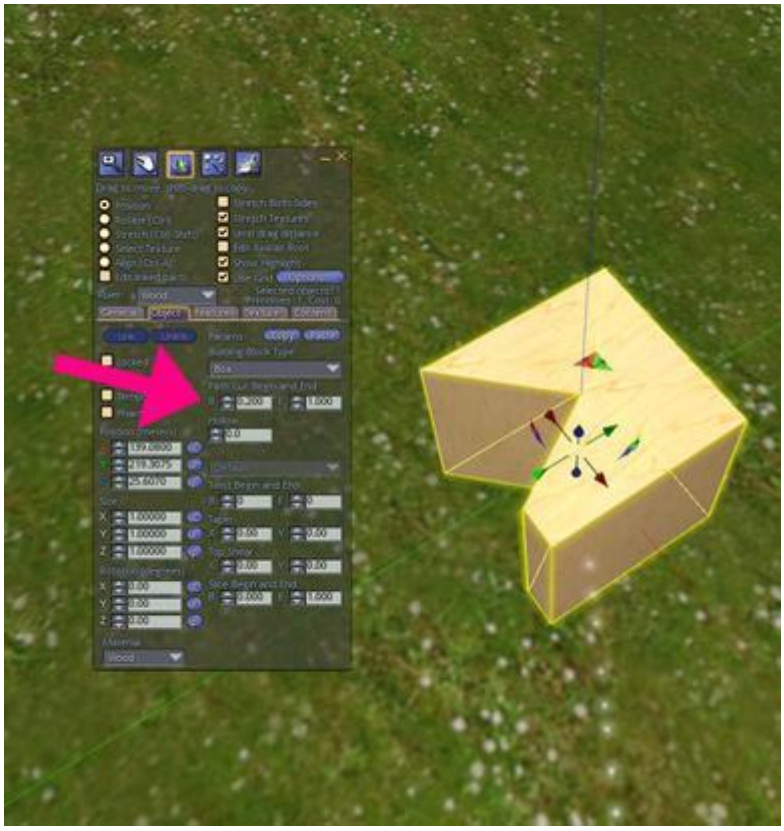
You can cut different sized chunks out of shapes using the Path Cut Begin and End control. You will experiment with different basic shapes to see how this control affects each of them. Let's begin!

Practice

Step 1: The Path Cut Begin and End control

The **Path Cut Begin and End control** is located below the Building Block Type dropdown menu. The Path Cut Begin and End control will cut a section out of any shape you rez. The path it cuts is along the Z axis (the blue axis). The width of the slice, or chunk, that is removed depends on the Begin (B) and End (E) number values you choose. Values can range from 0.000 - 1.000.

Rez a cube. Make sure your cube is in Edit mode. With the Object tab of the Edit window open, use the up/down arrows of the Path Cut control to change both the B (Begin) and E (End) number values.



Do you see how your choices affect the size of the "chunk" that is removed? Selecting a small value will look like a small slice has been cut out of your prim. A large value will leave only a slice.

Try removing slices or chunks from each of the other basic shapes. Experiment with different number settings to see if you can make a shape exactly the way you want it. Take your time!

Naming and saving favorite shapes is a real timesaver when you build. Good builders don't recreate a common shape each time they build; they take it out of their inventory. Take a few moments to rename and save some of the great new shapes you think you will use in your future building projects. You might also take a few minutes to practice your texturing. Slightly changing the color of a cut face can really improve the look of an object.

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Hollow and Hollow Shape

What if you could cut a hole in the middle of a shape and hollow it out. You can!

Instruction

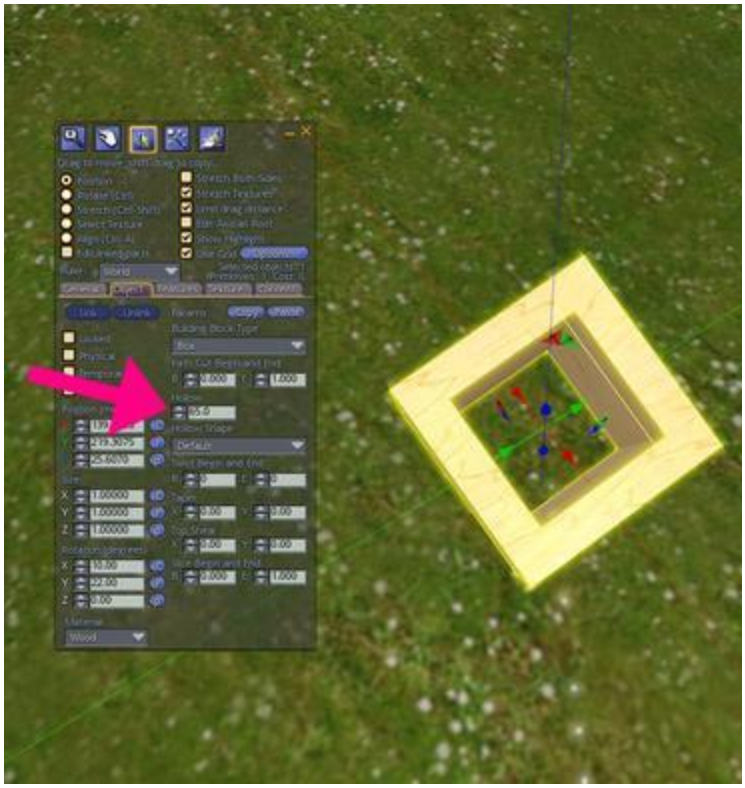
Every basic shape can be hollowed out. In fact, even the shape of the hole can be changed. For example, you can put a square hole in a sphere or a round hole in a pyramid.

Practice

Step 1: Making a default hole

When you first rez any basic shape, it is not hollow. To hollow a shape, choose the Hollow control, which is usually right underneath the Path Cut Begin and End control. A shape can be hollowed up to 0.95% in most viewers, though some viewers allow a higher percentage. It is usually hollowed along the Z axis (blue axis).

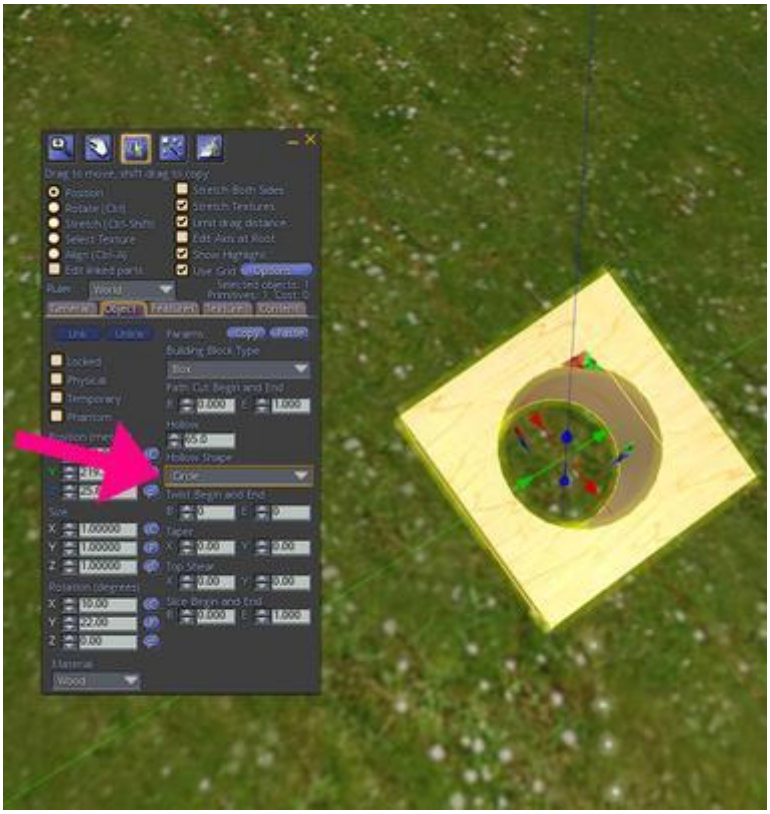
Rez a cube. Using the Hollow control, select the up/down arrows to hollow the cube. Do you see how the higher numbers make the hole in the cube larger?



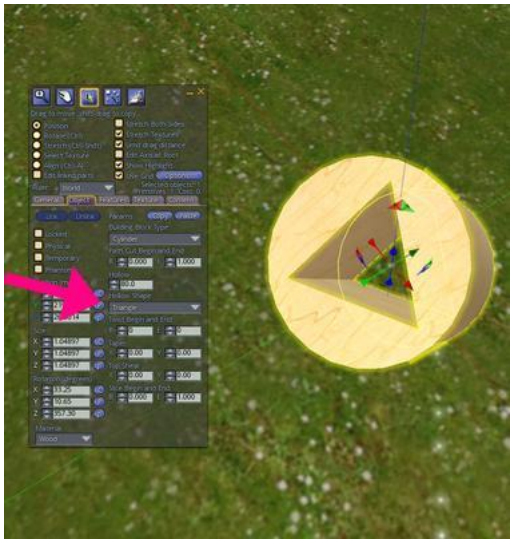
Step 2: Changing the default hole shape

Once you begin to make even a small hole in a shape, the Hollow Shape drop down menu will become active. Now you can change the shape of the hole to a circle, square or triangle.

Rez a cube and hollow it. Rotate it so you can see the shape of the hole. It should be square to match the square shape. Then select the Hollow Shape drop down menu and select Circle or Triangle. Notice how the shape of the hole changes instantly!



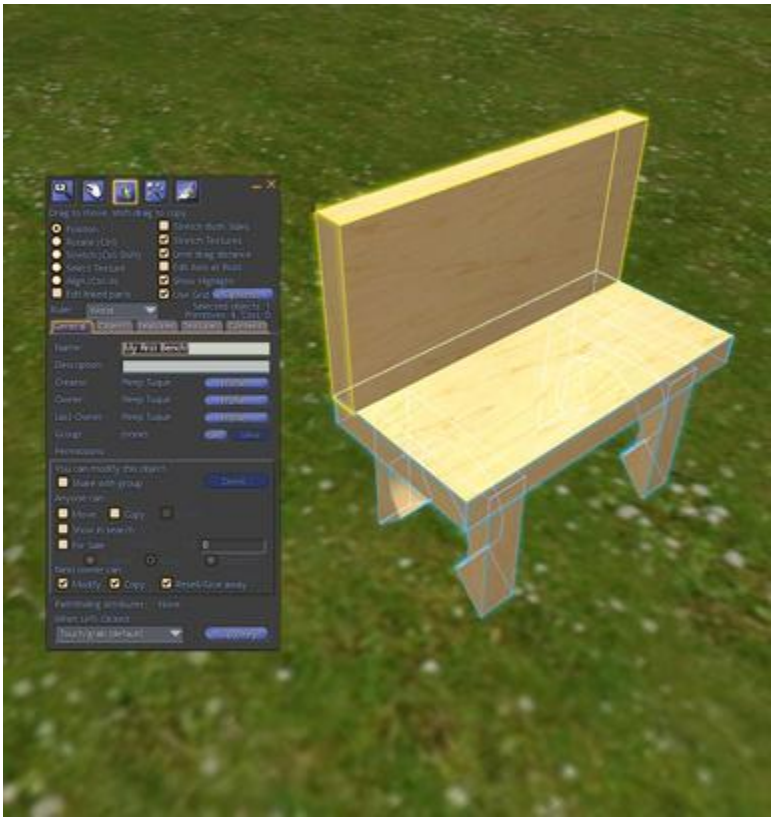
Step 3: Hollowing other shapes
Actually seeing what happens to other shapes can inspire you with new building ideas.



Select other shapes and experiment with hollowing them. Change the shape and size of the holes. Note: The Ring, Torus and Tube shapes hollow much differently than the other shapes. Make sure you take the

time to really look at them.

Step 4: Combining Hollowing with Cutting
When you begin to combine different building controls, you can really come up with some interesting shapes.



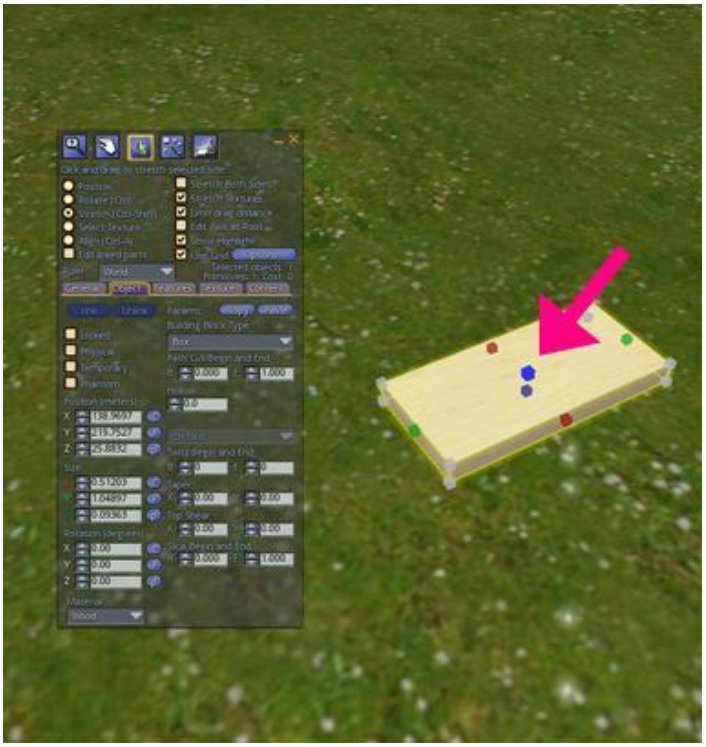
Look at the bench in the photo. It is made of three basic cubes. You've already learned everything you need to recreate it. Can you?

- Hint: You can find the steps below if you get stuck.

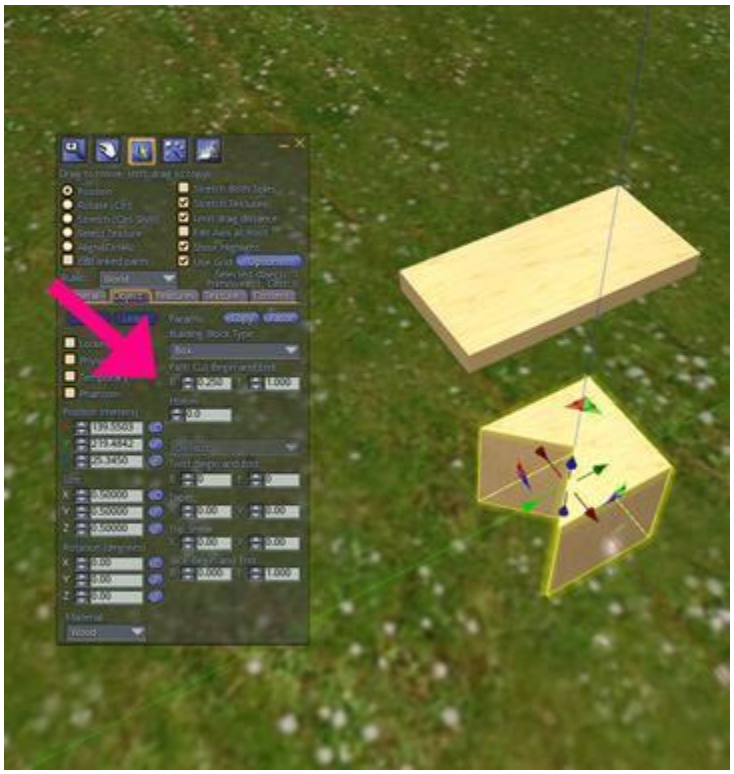
Using all the building knowledge you have so far, make something wonderful. You are going to be a fabulous builder!

Spoiler: How to make the bench

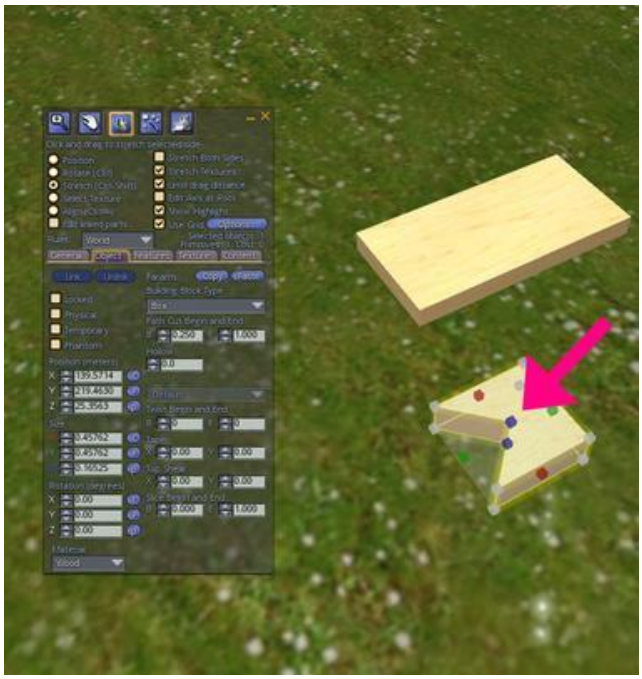
Rez a basic cube. Stretch and flatten it to make the bench seat.



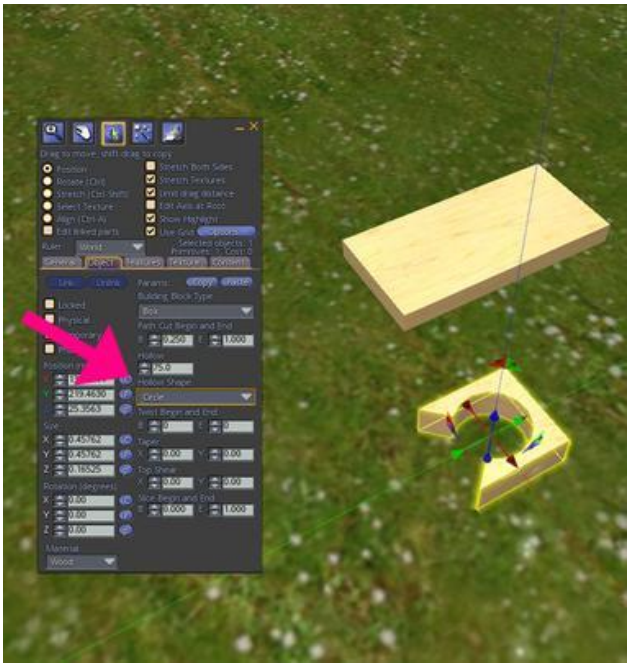
Rez a second cube, and cut it to make a "V" shape for the legs.



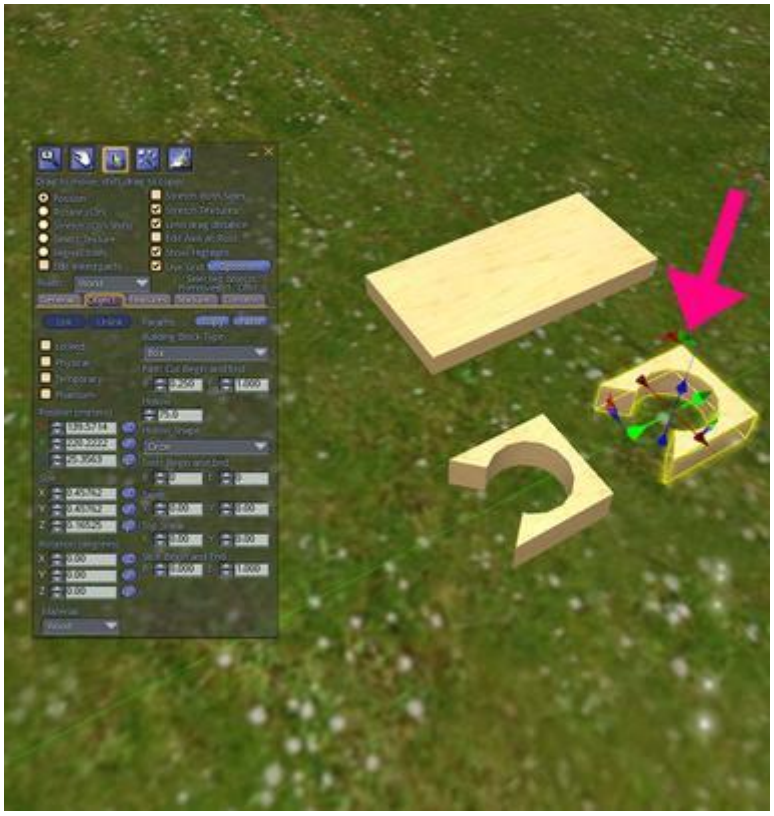
Narrow the width of the legs to your liking by squishing it down along the Z axis (blue handle).



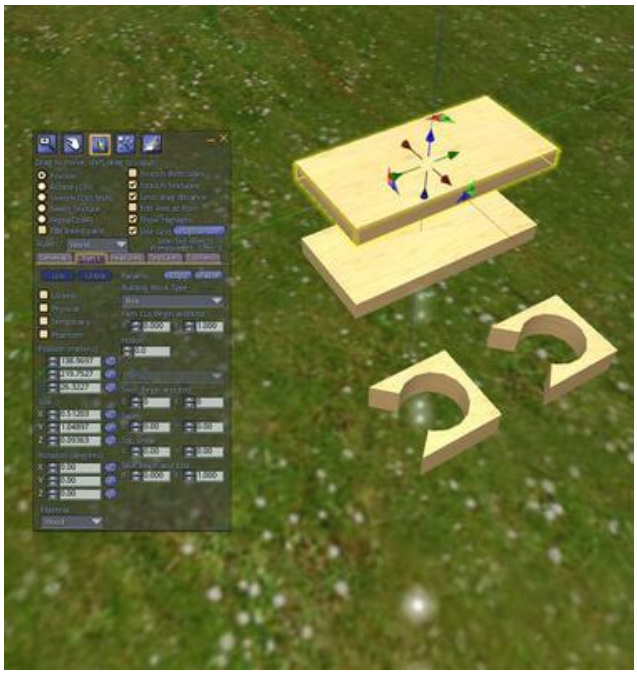
Now hollow the second cube to your liking and change the Hollow Shape to circle. (You may need to rotate it at this point.)



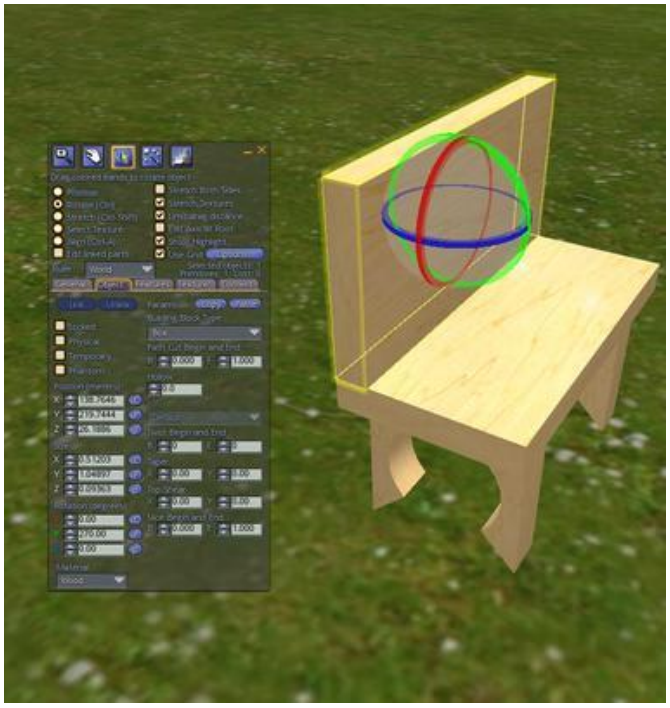
Select the leg section and hold down the **SHIFT key**, then drag along one of the arrows to make another leg section that is identical to the first. This is called the "shift-drag" method of copying.



Also duplicate the bench seat to form a prim for the top of the bench seat by holding down the SHIFT key and dragging up on the Z Axis (blue arrow).



Now, using rotation (CTRL) and moving the prims with the arrows, position the legs and top so they look like a bench.



Link the four prims together (hold down Shift key to highlight all parts, then CTRL+L to link) and rename. Nice job!

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Taper, Top Shear and Dimple

You have already come a long way in learning how to edit and change basic prim shapes. Now things are about to get really interesting. Different shapes can react much differently to the same control. Two of those controls are Taper and Top Shear.

Instruction

When you taper something, you usually make it smaller at one end. When you shear something, it usually looks a little lopsided - like it's leaning over a bit. However, how Taper and Top Shear look with different basic shapes can be very different! For this mission, your job is to really experiment with all the different basic shapes to see what happens. Note: As you follow along, you will notice that different shapes show Taper and Top Shear in different places in the Object tab of the Edit window.

Practice

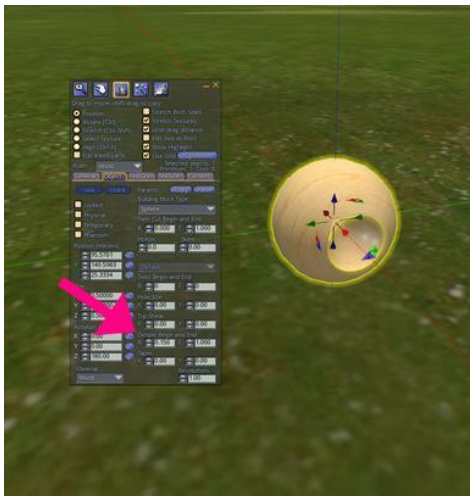
Step 1: Taper
When a prim is tapered, the top end usually becomes smaller. A prim can be tapered from 0.00 to 1.00 percent on both the X (red) and Y (green) axes. If you can choose negative taper values, such as -0.01 to -1.00, [for a prim, the bottom end will be tapered.

Rez a cube and a cylinder. For both, use the Taper control up/down arrows to taper both the X and Y axes to 1.00. Did you get a pyramid and a cone? Experiment to see what happens if you use smaller taper values.

Step 2: Tapering other shapes
While a sphere cannot be tapered, the other shapes can. The results can be really surprising!

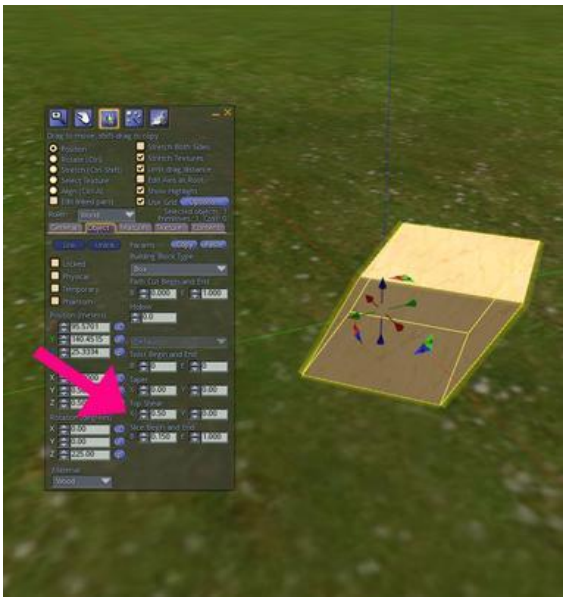
Rez all the other shapes (except for the sphere and half sphere). Really experiment with the amounts of X and Y taper. What is the most surprising shape you have discovered so far?

Step 3: Dimple
Spheres and half spheres cannot be tapered. Instead, they can be dimpled Just like the dimples on someone's chin, a dimple on a sphere makes a small dent, just at the end.



Rez a sphere. Change the numbers in the Dimple Begin and End control to see what happens. What might you make if you also hollowed that sphere?

Step 4: Top Shear
For some prims, such as cubes and cylinders, the Top Shear control will make them look like they are leaning to one side. For other prims, such as toruses, or rings, Top Shear will distort them in unexpected ways! You can use Top Shear on the X (red) or Y (green) axis - or both!



Rez each basic prim shape and experiment with the Top Shear control. Change the settings for both the X an Y axes and see what happens.

Are you beginning to see how much you can change those basic prims? It can really be fascinating to play with shapes. At this point, you are probably surrounded by your strangely-shaped objects. Take a moment to clean up. Name and save shapes you really like and delete the rest.

Now travel around the OpenSimulator Hypergrid, and R-click (Cmd-click) on objects you see and choose Edit. The prims will be outlined and you can see what kinds of shapes are being used in some of your favorite builds!

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Twisting Prims

When you began this module, you may never have dreamed you could make these kinds of shapes. But wait - there's more! You can twist prims, too.

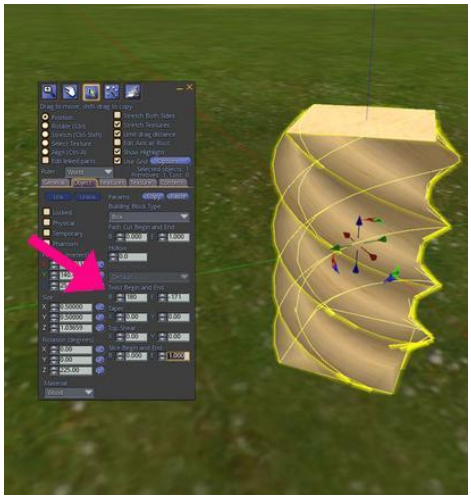
Instruction

In this last section on shapes, you will learn to twist different shapes in different ways. Much like twisting a towel or crumpling a piece of paper, twisting prims will lead to some very interesting shapes. Take your time and fully explore how the "twisting" controls affect each different shape. For some shapes, you will have additional twist controls, which will be explained along the way.

Practice

Step 1: Simple Twist

Some shapes, such as cubes and cylinders, have only a simple Twist Begin and End control. To really twist these shapes, choose a large beginning number and a large negative ending number.



Rez both a cube and a cylinder and experiment with twisting them using the Twist Begin and End control. To really see the twists, you may want to stretch them into columns first. Other shapes that are based on the cube and cylinder shape will act much the same way. Try twisting pyramids, triangles and cones.

Step 2: Twisting Spheres
Like cubes and cylinders, spheres have only the Twist Begin and End control. However the results of using that one control are really neat!

Rez a sphere. Change the numbers on the Twist Begin and End control to see what happens. Pretty fascinating!
Tip: Combining the control with hollow and dimple can make amazing flower petal-like shapes.

Step 3: Toruses, Tubes and Rings
As you've seen all along, these shapes always seem to do unexpected and surprising things. Twisting them is no exception. You can easily make springs and curl shapes. In fact, these shapes have additional "twist" controls you can change. In addition to Twist Begin and End, you can control the:

- Hole Size - The size of the hole in the middle of the spring.
- Radius Delta - The extent to which each individual coil becomes smaller or larger.
- Revolutions - How many coils are in a spring shape.
- Skew - How flat or rounded each coil appears.

Play with these and you will end up with some amazingly complex shapes. This is one reason that many plant-like and hair prims begin with toruses or tubes - they can be changed in radical ways.

Rez a torus. Begin by changing the Twist Begin and End controls. Now, experiment with the Hole Size. Do you see the size of the hole in the middle of the spring changing? Finally, change the Radius Delta and Revolutions controls to see what they will do. As you change the Radius Delta, each individual coil in your spring will get smaller and smaller, like a pyramid of coils. The higher the Revolutions, the more coils in your spring!

Tip: Expert builders spend a lot of time learning how to make exactly the "right" shape. There is just no shortcut to figuring these shapes out. If you find you've made something unique and beautiful, name and save it!

Learning about shapes never ends. Just when you think you've seen it all, you learn something new. The best builders never stop learning! What would happen if you were to combine "twists" with all the other techniques you've learned in this module?

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Additional Features (Parameters)

OVERVIEW
You are now well on your way to creating whatever you can dream up! There are just few more building features that can make your creations really pop. Did you know that you can make a prim wiggle like jello or light up the area around it? You can also create a prim that can be walked through, or can fall to the ground or even disappear after a short time.

Learning Objectives:
By the end of this module, you will have the following skills:

- Changing qualities of objects so that they can disappear, be walked through or fall to the ground
- Making objects that are flexible
- Modifying objects so that they give off light
- Using sculpted prims

You will demonstrate your new skills by:

- Creating a variety of objects that wiggle, flop, light up and even disappear!
- Rezzing and modifying complex prims (sculpted prims)

Parameters

Have you ever wished you could fly through the roof or walk through a wall? What about making an object that just disappears after a minute? You can do all that and more just by selecting some easy-to-use building features called parameters.

Instruction

In OpenSimulator building, a parameter defines a quality or characteristic of a prim. You can set a prim so it cannot be moved. You can also set it so that it falls to the ground, can be walked through or disappears. Let's practice these easy features now.

Practice

Step 1: Finding the object parameter controls
The object parameter controls are located in the Object tab. Edit Window > Object Tab > Object Parameters

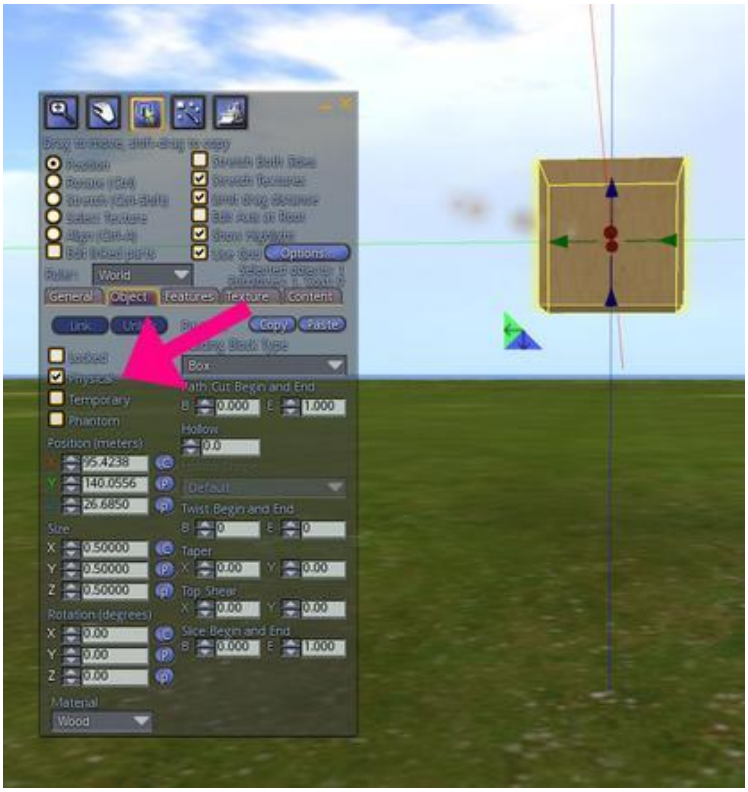
Find the object parameters. Notice that they are unchecked.

Step 2: Locking an object
Choosing the Locked parameter prevents an object from being accidentally moved or changed while building. It's very useful when building a complex object or when building with others.

Rez a prim. Choose Locked from the object parameters. Try to move it!

Step 3: Making an object physical
Selecting the Physical parameter makes an object that looks like it responds to gravity. If it is rezzed in the air, it will fall to the ground!

Rez a prim and choose Physical from the object parameters. Position it so that it is in the air above the ground. Close the Edit window and watch it fall!



Step 4: Making an object temporary
A Temporary object will disappear one minute after it rezzes. This feature can be great for special effects, like fireworks, or when you want to quickly show someone an object. You could even use a temporary prim as a stopwatch!

Rez a prim and check the Temporary parameter. After one minute, you should see it disappear.

Step 5: Making an object phantom
An object that is Phantom can be moved through. If a wall is Phantom, you can walk through it!

Rez a prim and stretch it so it's larger than your avatar. Select Phantom from the object parameters. Now walk or fly through your prim!

These parameters are easy to use. You can even combine them! Could any of your previous creations use these features? Go modify them!

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Flexi Prims

Flapping flags and swirling skirts. What do these have in common? Flexi prims!

Instruction

Flexi prims are regular prims that have flexible characteristics. They can seem to droop with gravity or sway with the wind. The flexi controls are easy to use and experiment with. Let's get started!

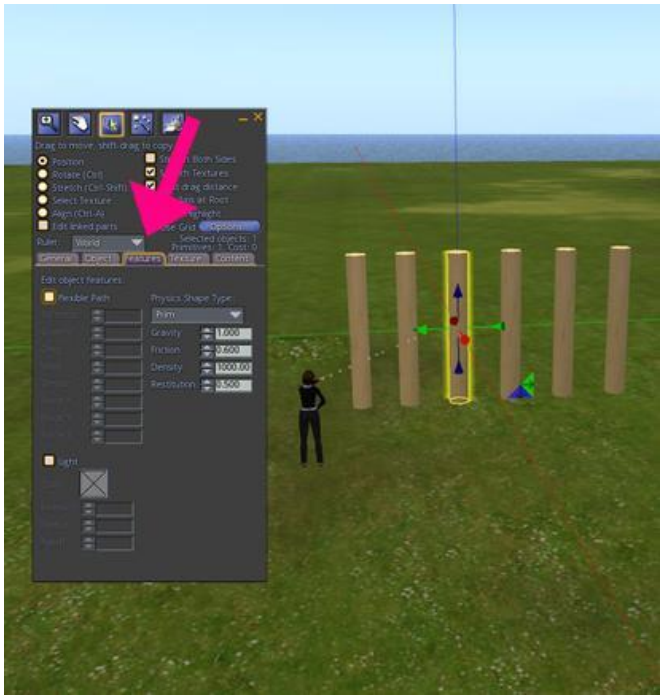
Practice

Step 1: What is a flexi prim?
Flexi prims:

- Can be made from any prim based on a cube or cylinder shape.
- Are always phantom.
- Cannot be physical.

Rez at least six cube or cylinder based shapes. Stretch each one so that it is taller than your avatar. You will change each one as you explore the different ways an object can become flexible!

The Flexi controls are located in the Edit window in the Features tab. Edit Window > Features Tab > Flexible Path Checking the Flexible Path box will make a prim flexible.



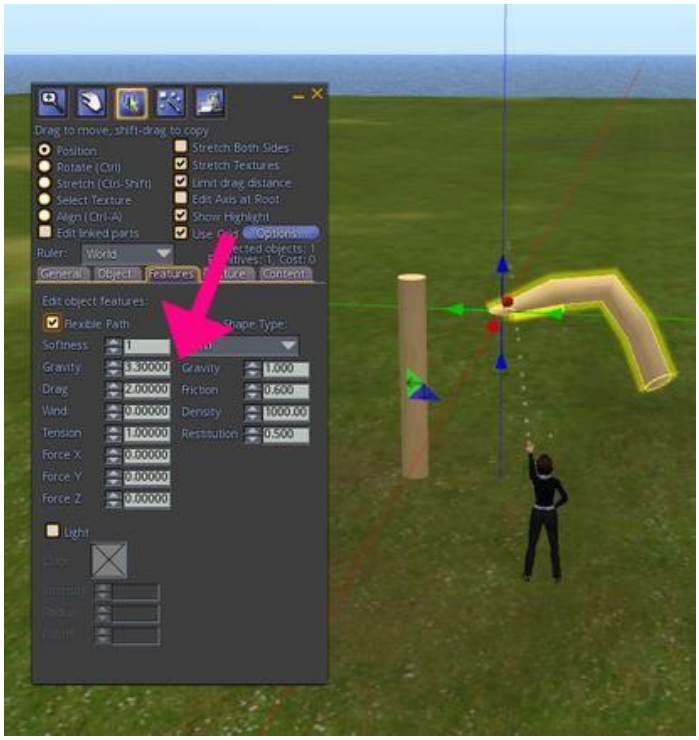
Edit a prim to make it flexible. You will see each flexible feature is preset with specific numbers.

Step 2: Softness
The Softness feature controls how soft, or floppy, a prim will appear.

Edit one of your prims and make it flexible. Practice changing the level of Softness. The higher the number, the softer the prim. Move your prim around to really see how soft it looks.

Step 3: Gravity
This control determines how much a prim will respond to gravity. A high number will make your prim slump over. You can even choose negative numbers if you want an object to have mysterious anti-gravity qualities!

Edit another prim and practice changing only the Gravity setting.



Step 4: Drag
Drag is the amount of air friction on an object. An object with low drag will wiggle like crazy! A high amount of drag will allow an object to sway gently.

Edit a prim and change the amount of Drag.

Step 5: Wind

The Wind settings determine how much a prim will react to the wind. Picture a wind sock or flag that changes direction with the wind. Edit a prim and change the Wind settings. Try rotating and changing the position of the prim to see how it reacts to the wind.

Step 6: Tension

The Tension control adjusts the amount of springiness in a flexi prim. The lower the number, the springier it will be. Change the amount of Tension in a prim. You may need to move it around to notice the change.

Step 7: Force X, Y and Z
Remember the axis directions? Red: X axis - Green: Y axis - Blue: Z axis Increasing the Force of an axis makes the prim flop along that axis.

Edit a prim and change the amount of Force along each axis. Notice how the prim aligns itself along an axis as you increase the number.

There are many ways to bring your scenes to life with the addition of simple flexi prims. Experiment with the settings to make a terrific flag!

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Adding Light

You already know how to make an object glow from within using the Fullbright feature. What if you could make an object give off light like a light bulb? The Light feature allows you to do that!

Instruction

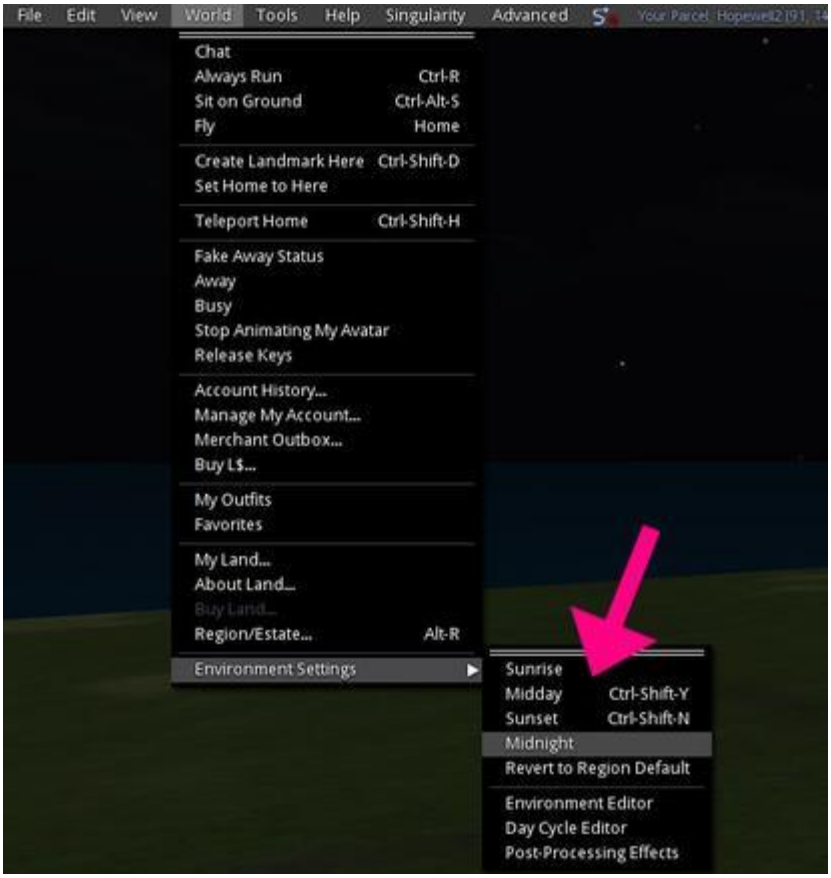
You can set an object to emit, or give off, light by using the Light feature. Just like a light bulb, it can be dim or bright.

Practice

Step 1: Changing day to night

To really see light effects, the time of day should be Midnight. You can force the time of day on the region you are in by changing the Sun settings. Where these settings will be depends on what viewer you are using. In some viewers, the setting is in the World menu, World > Force Sun > Midnight. Other viewers call the menu Environment Settings, World > Environment Settings > Midnight. You will have to poke around to find the Sun settings on your viewer.

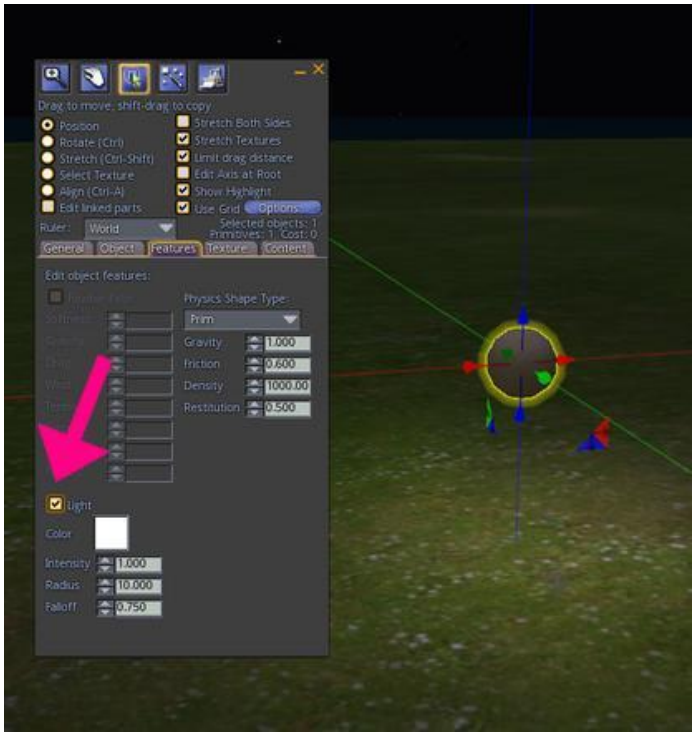
Remember, changing the day/night time on your viewer does NOT change the time for anyone else!



When you are done with this section, you can use the same feature to change the sun back using Revert to Region Default.

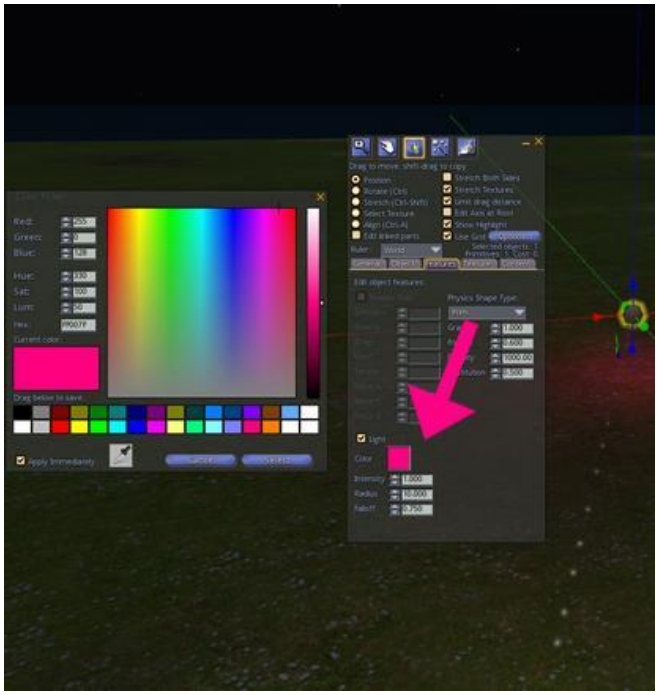
Step 2: Making a prim have light

The Light controls are located in the Edit window in the Features tab. Edit Window > Features Tab > Light Checking the Light box will make a prim emit light. Rez a prim and select Light. You will see each feature is preset with specific numbers.



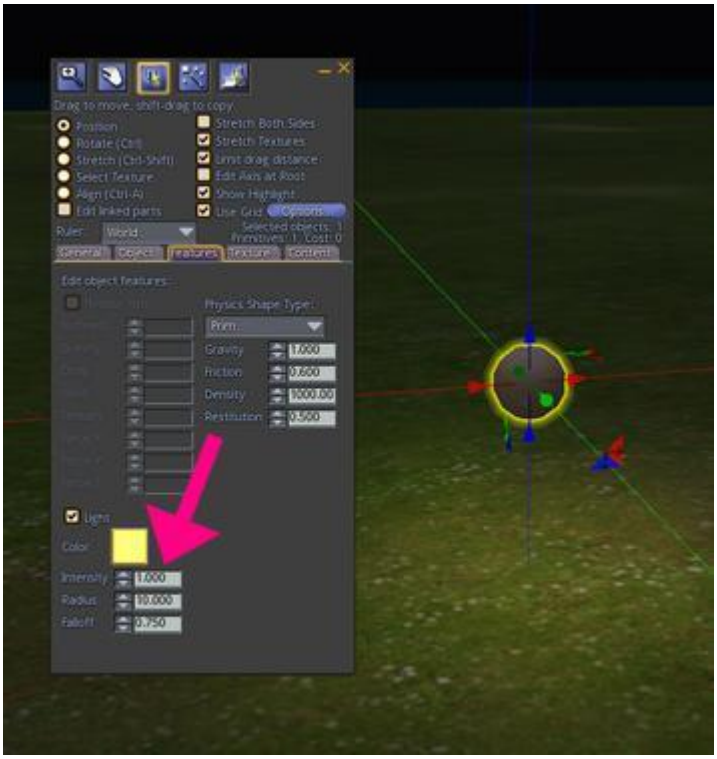
Step 3: Setting a light's color
Changing a light's color will not change the color of the prim, only the light it is emitting. To change a light's color, click on the white box to bring up the Color Picker window.

Change the color of the light coming out of the prim you rezzed in **Step 2**.



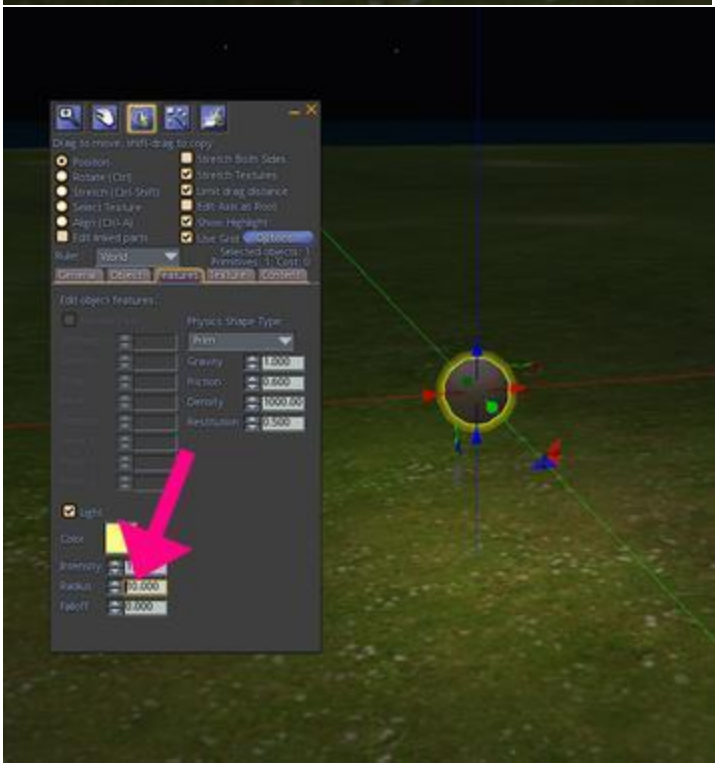
Step 4: Setting a light's intensity
A light's Intensity refers to how brightly it shines on other objects nearby. The higher the number, the more intense its light.

Change the light intensity of a prim.



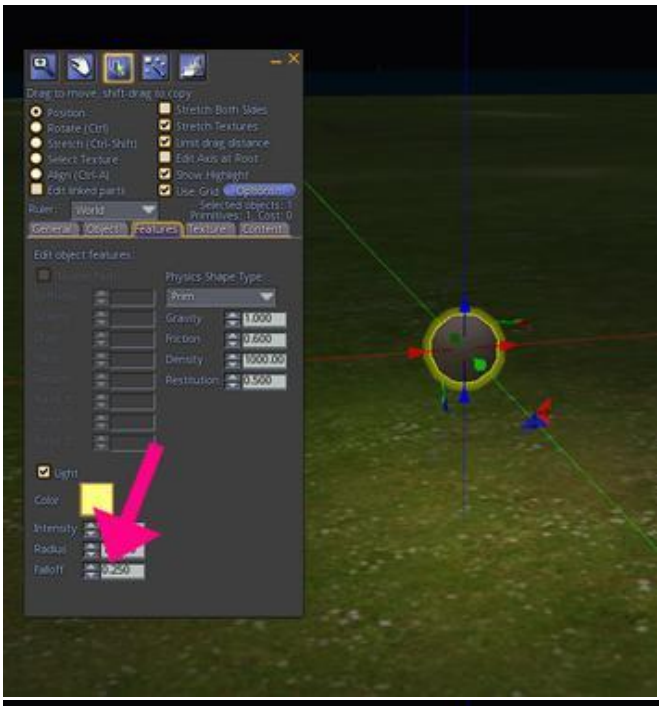
Step 5: Setting a light's radius
A light's Radius is the distance a light shines. A low number will cast light for a small area such as around a campfire. A high number allows the light to travel as much as 20 meters.

Change the light radius of a prim.



Step 6: Setting a light's falloff
Falloff refers to how fuzilzy or sharply a light ends. A light with a low Falloff number will very gradually fade away.

Change the light falloff of a prim. You will be able to notice this effect much more if the light also has a high intensity. Tip: Although you can set how far a light can travel, you cannot set its direction. You can use Radius and Falloff to make sure your light effect stays just in the area you want it to.



Adding subtle light effects can add drama and mystery to an object. Try setting out a few lights and see

how the look of an object changes - especially at night.

Tip: You can add lights even if the prim itself is transparent!

One other word of advice: How light displays for each person in the area will depend greatly upon their individual viewer settings, so be careful of using TOO MUCH light in an area - it can wash out the scene for users with high quality video cards. It's best to use a light here and a light there, but not tons of lights in a scene. In addition, the viewer can only display so many lights in a given area, so it's best to use light sparingly.

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Texturing

In addition to learning to build with prims, you also need to learn about **texturing** the shapes to create the "skin" on the outside of each shape.

For example, a large vertical rectangle-shaped prim could be an interior wall if you apply a wallpaper texture or it could be part of an exterior wall if you use a brick texture.

The basic shape of the prim doesn't change, but the look and feel of the object will vary depending on the textures used during construction.

Although building a good model or shape is a critical skill, the single most important aspect of creating a space that looks realistic or professional is the application of textures to the shape. A great model can be ruined by bad textures, and even a simple model can look fantastic with a good shaded texture!

In this chapter, you will learn:

- An Introduction to Textures
    - Color and Transparency
    - Uploading and Applying Texture Images
    - Repeats
    - Offsets
    - Fixing Flicker
- Advanced Texturing
    - Shiny and Bumpy
    - Light and Shadows
    - Using Textures Wisely and Well

Move on to the next page to begin your texture adventure!

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

An Introduction to Textures

OVERVIEW
As you look around the virtual world, you will notice that objects are not only made up of shapes, but also the patterns visible on the outside of the shapes, known as ***textures***. Great-looking objects have great-looking textures! In this section, you will learn how to apply great-looking textures to everything you build.

TERMS:
Texture: A graphic or image applied to an avatar or object.

Learning Objectives:
By the end of this module, you will have the following skills:

- Putting graphics and images (textures) onto objects you create
- Changing the color and transparency of an object
- Changing how the size of a texture looks on an object
- How to upload a texture and apply it to your objects

You will demonstrate your new skills by:

- Texturing created objects
- Tinting and changing the transparency of created objects
- Changing a texture to look the right size on each side of an object

Finding and Using Textures

What makes something look like it's made of stone, fur or metal? The answer: Textures. Textures are graphics or images that are applied to an avatar or object. Now that you have some basic building skills under your belt, it's time to make your creations look like something other than plywood!

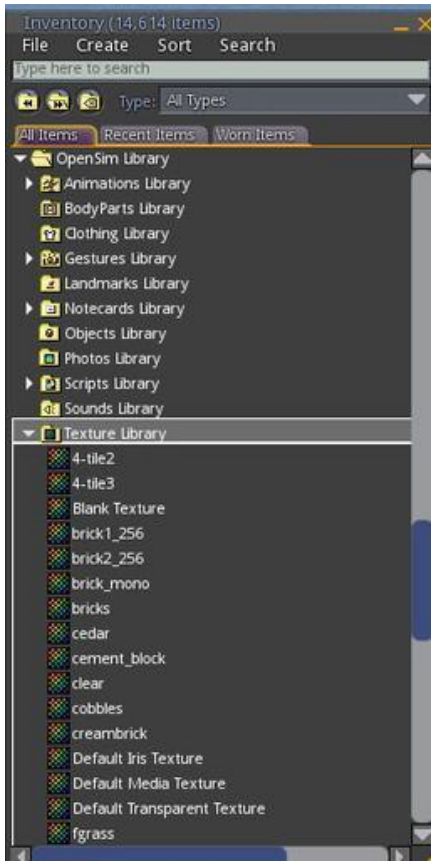
Instruction

Applying textures to an object is quick, easy and a ton of fun. In this section, you will learn how to find images and textures and place them on an object.

Practice

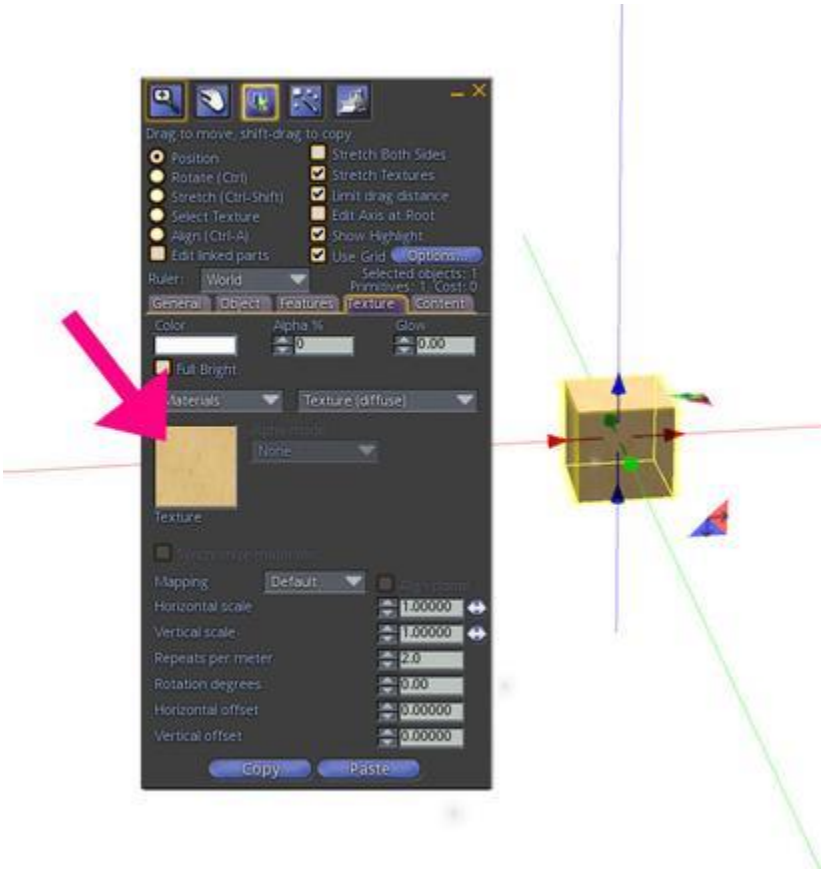
Step 1: Finding textures

You can use both Textures and Snapshots to change the look of your object. You have been given some textures and snapshots to use in your Inventory Library. Inventory >OpenSim Library > Photo Album or Textures

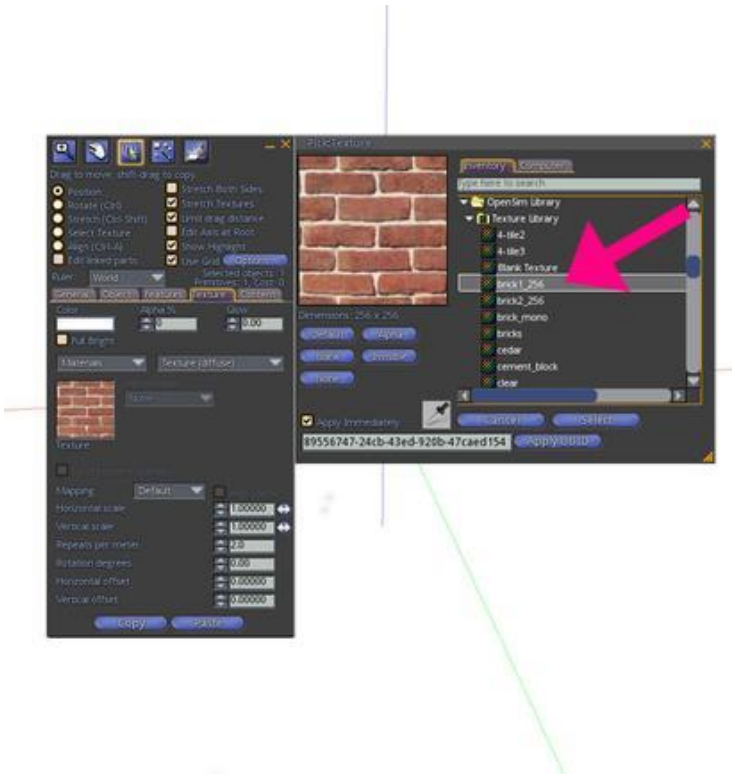


Find the Photo Album and Textures folders in your Inventory Library. Open the folders and double-click on a few textures and snapshots to see what you have there.

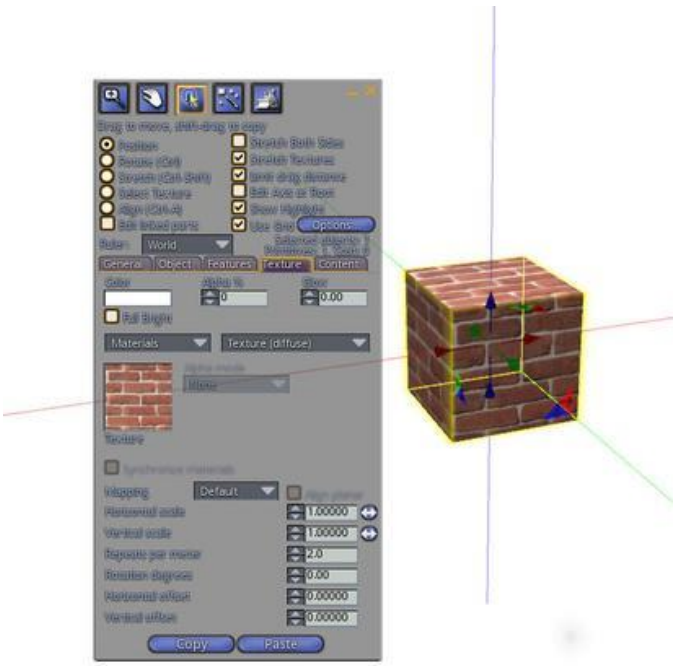
Step 2: Putting a texture on an object
Rez a cube and make sure it's in Edit mode. Choose the Texture tab. You will see a small square in the Texture tab labeled Texture. This is the texture that's on your prim. Plywood, right?



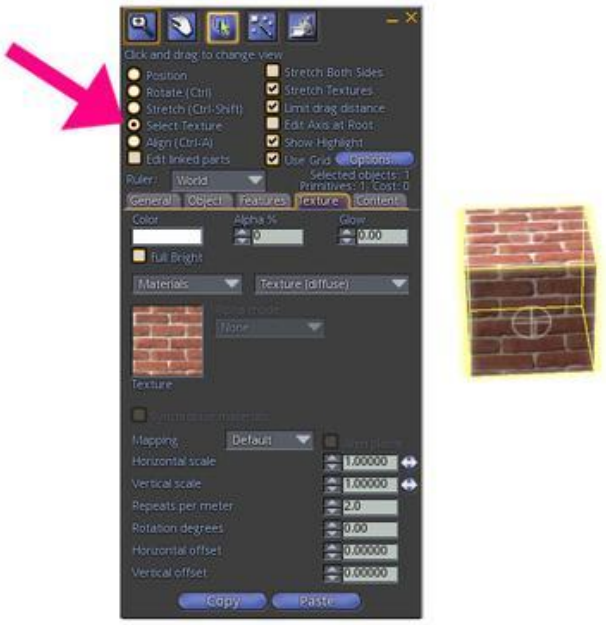
To change that texture, single or double-click (depending upon your viewer) on the plywood square in the Texture tab. The Texture Picker window will come up, showing the texture currently on your object. It will also show you a list of all the textures and snapshots you have in your Inventory.



Double-click on the texture square in the Texture tab. In the Texture Picker window, navigate to: OpenSim Library > Texture Library > brick1_256 to apply it to your prim. Did you see your prim change? Note: You can also search for a specific texture using the Search bar at the top of the Texture Picker.



Step 3: Changing the texture on one side of a prim
Each *face* (side) of a prim can have a different texture. To change the texture of each face, choose the Select Texture button from the Edit window. You will see a target on each face of the prim.



Click on the face you would like to change. Now, you may select a new texture by double-clicking on the Texture square in the Texture table and clicking on a texture from the Texture Picker window.

Select each face of your prim and apply a different texture to each one. (To see the top and bottom, you may need to reposition your prim or use Camera Controls for a better look.)

Using your new building skills, take some time to rez some different shapes and sizes - and then texture them. Notice how the same texture looks on different shapes.

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Color and Transparency

Sometimes a texture is almost perfect. If only you could change the color just a bit. Guess what? You can. In fact, you can even make it almost transparent for special effects.

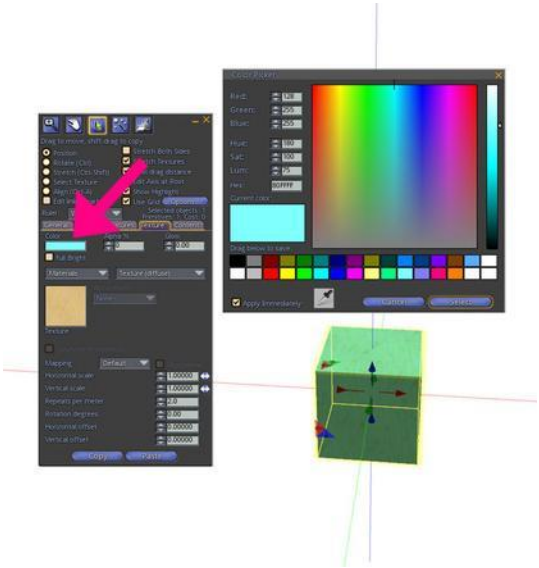
Instruction

While you can't completely change colors in a texture, you can tint a texture or make it a solid color.

Practice

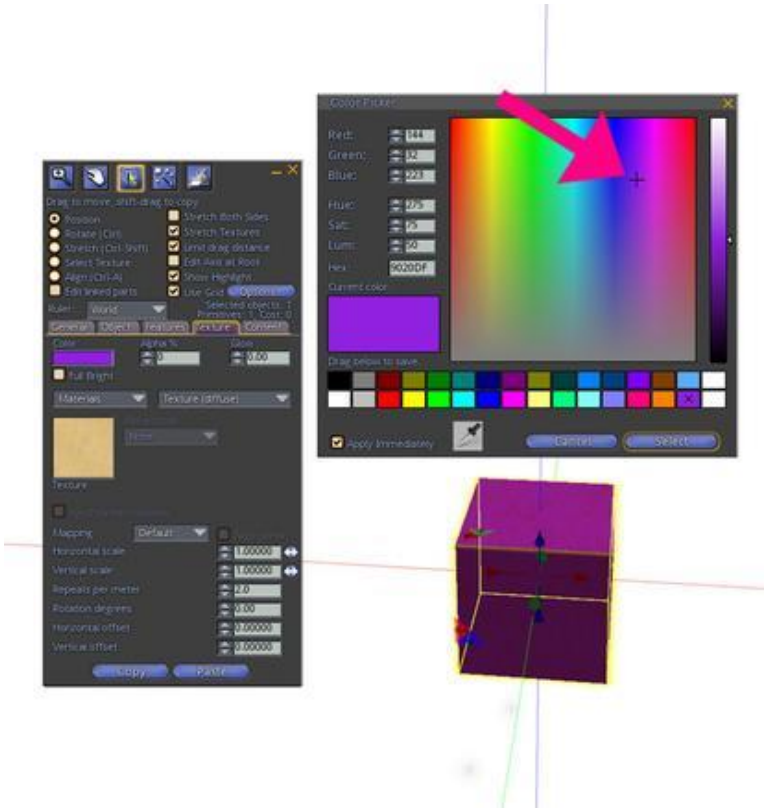
Step 1: Tinting a texture

Rez a prim with the default plywood texture. Make sure it is in Edit mode. In the Texture tab, right next to the Texture Picker window, is a white box called Color. This is the Color Picker window. Click on it to bring up the Color Picker. By default, the Apply Immediately box in the lower left corner should be checked. To change the tint of your prim, click on any of the colored boxes towards the bottom of the Color Picker.



With your prim in Edit mode, choose the Color Picker and click on one of the colors already provided for you. **Notice that the wood grain still shows through the tint.** Coloring a texture does not replace the texture, it merely *tints* the texture with the color you've chosen. Be careful, sometimes colors don't mix well with an existing texture. It's best to tint textures within the same color family unless you intend to radically change the color.

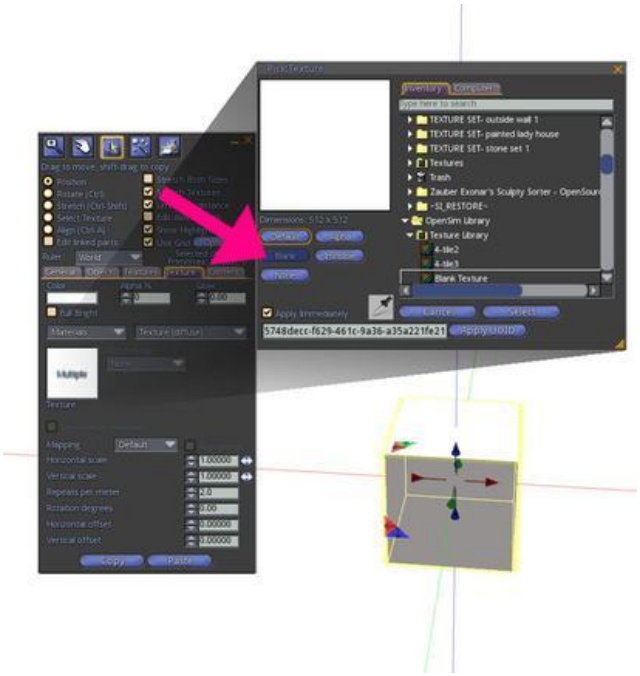
Step 2: Choosing your own colors
You are not limited to the colors already in the Color Picker. You may also design you own and save them to use on your projects. In the Color Picker, click anywhere on the large rainbow of colors (Spectrum). The exact color you've chosen will appear in the box Current color. You will also see a slider to the right to adjust your color to be lighter or darker. If Apply Immediately is checked, you should see your prim take on the new color right away. If you would like to save that new color, just drag it out of the Current Color box to replace one of the default colors.



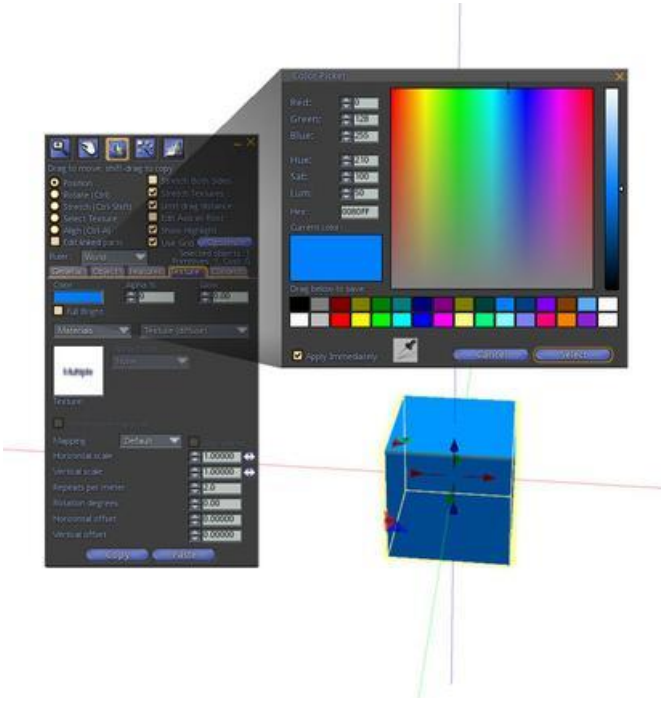
With your prim selected, open the Color Picker and use the Spectrum to create a new color. Click on the light/dark slider to adjust. Drag it from the Current Color box to replace a default color. Note: You may also choose RGB values in the upper left corner to select a color.

Step 3: Choosing a solid color

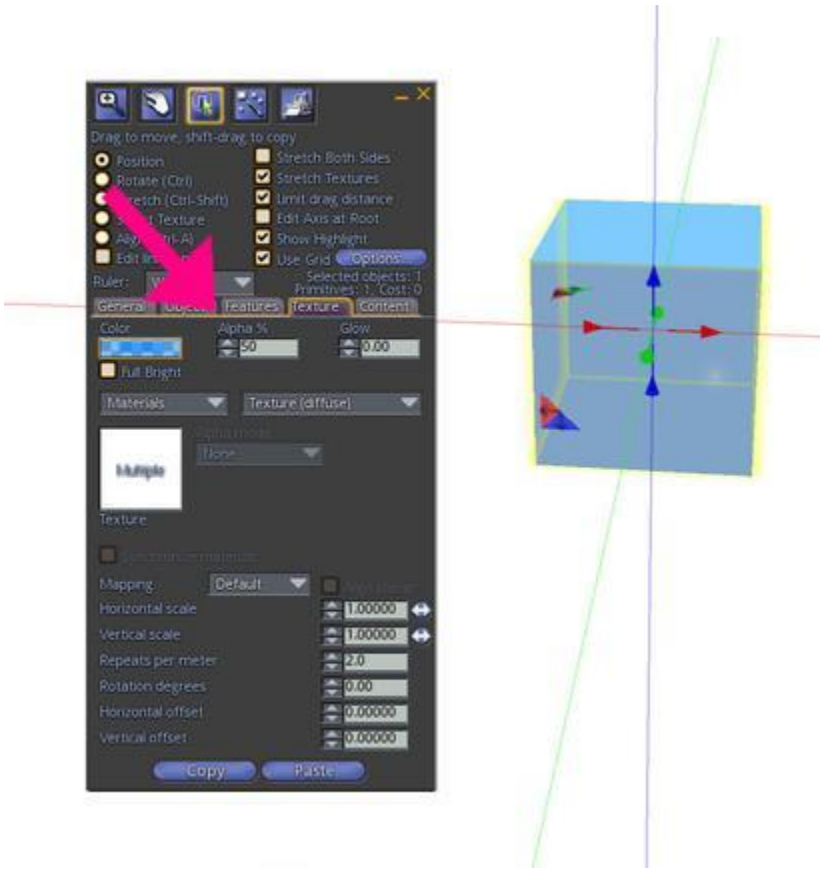
To make a prim a solid color, open the Texture Picker and choose the Blank button. Your prim will look solid white.



Now, you can use the Color Picker to tint it a solid color. Note: You may also select the Select Texture button to tint each face of a prim separately.



Step 4: Transparency
You can make an object up to 90% transparent using the Transparency or Alpha control (depending upon your viewer, it may be labeled one or the other). While 0% is fully visible, 90% will make your object look like a faint shadow.



Change the transparency of your prim. Note: There are completely, 100% transparent textures to be found in the OpenSim Library, but be careful! Once you make a prim completely transparent, it can be difficult to find it again.

Try this for fun: Rez a cube and stretch it to look like a wall. Texture it with something like bricks or stone. Now, use Select Texture to select just one side and change the transparency to 90%. If you look at it from all sides, you will find you just made a one-way see-through wall!

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Uploading and Applying A Texture

What if you could use a pattern or image you have on your own computer to texture an object? You can! You can upload images that are in a JPEG (JPG), TGA or PNG format. You cannot upload GIFs. Usually, valid images will have .jpg, .tga or .png at the end of their file names.

!! A Special Note About Copyright & Image Permissions !!

It is absolutely crucial that you understand the legal implications of using an image you do not have the right to use on your creations in virtual worlds. By uploading an image that is copyrighted or owned by someone else and using it in your builds, you are exposing yourself, your team, and the owner or operator of the grid to legal liability from the legal copyright owner. **Under no circumstances are you permitted to use images or textures that are copyrighted by someone else unless you have their legal, written permission to do so!** Failure to comply may result in disciplinary actions up to and including termination if you are an employee.

Responsible builders understand that their creations must be fully compliant with the law regarding copyright. **If you are not 100% sure of the source of an image do not use it in your builds.** There are millions and millions of images that are available in the public domain or under a Creative Commons license, so there's no reason to steal an image when there are so many perfectly legal textures available. Also be careful using Creative Commons licensed images! The best images to use are released under a Creative Commons 0 license ("CC0"), which has no restrictions on their use. If a Creative Commons license requires attribution or that it only be used for non-commercial purposes, then you must be sure that your item and any*derivative* items created from your item all meet the license requirements. Attribution means you must state where the source of the image came from. Non-commercial use means that the image can never be used in an item that will be sold for money. This can be very tricky if you share your work with other people, so it's best to only use public domain or CC0 licensed items so you never have to worry about copyright in the future.

Instruction

OpenSimulator has the ability to upload - or copy - images from your own computer to be used as textures for objects, avatars and clothing. Even if you don't have any special imaging software on your computer, you can still upload photos you take or Internet images you have permission to use.

Practice

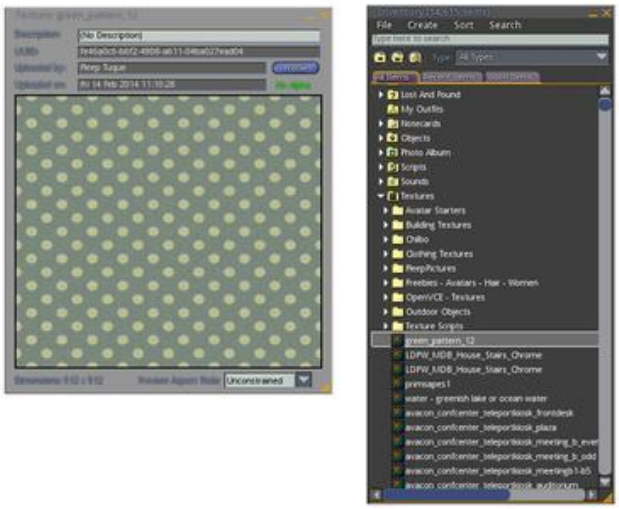
Step 1. Find an image to upload

On your own personal computer, find an image that you would like to import. Make sure you have permission to import it and that it is in the correct format (.jpg, .tga, or .png). Example images can also be downloaded from the Appendix "Sample Candy Textures" page. This will be especially useful for those using the Primland Building Tutorial.

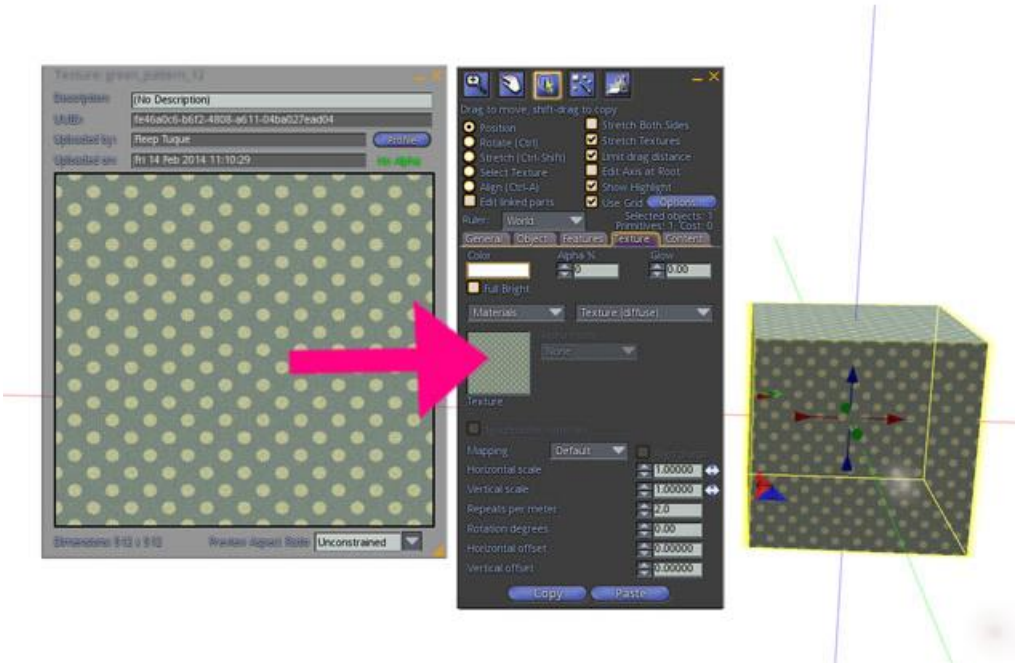
Step 2. Upload the image
In most viewers, you upload a file by using the File menu at the top of the screen. Under File, choose Upload Image. You will be able to search through your computer's folders and files to find the image you want.

In the Upload Preview window that opens, you will see the image you selected. You can rename it at this time. Some viewers offer a temporary upload option to temporarily preview your texture before uploading permanently. In most cases, you will want to choose the permanent upload button to continue or Cancel if you've changed your mind.

Once your image is uploaded you will see it open on your screen as a texture, and the texture will be in your Inventory > Textures folder for future use.



Now rez a prim and in the edit window > Textures tab, drag your new texture onto the texture box to change the texture of the object. Pretty great!



One other note, while technically you CAN drag a texture directly onto the face an object, in general this is a **bad idea** because it is very easy to accidently drag the texture onto the wrong object, which you may not have the permission or ability to repair. It is best to **always drag a texture onto the texture box in the edit window** to be safe.

In addition to free textures on the web, you can also use a digital camera to take photos of great textures you see - like a great stone wall, a mossy rock or a piece of old paper. And if you are skilled using an image editor, such as Photoshop, you can make great textures.

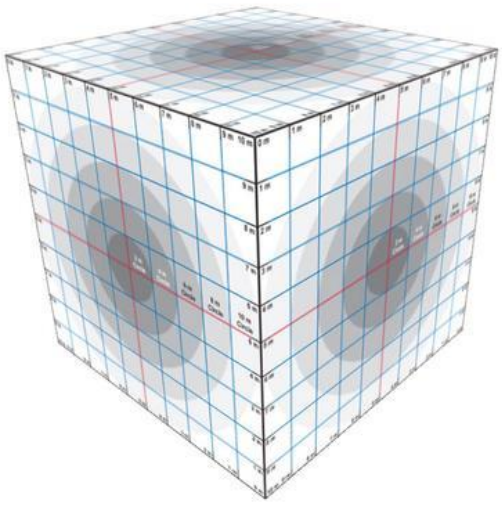
Do you have another image you would like to upload? Take a minute and do that now.

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Repeats

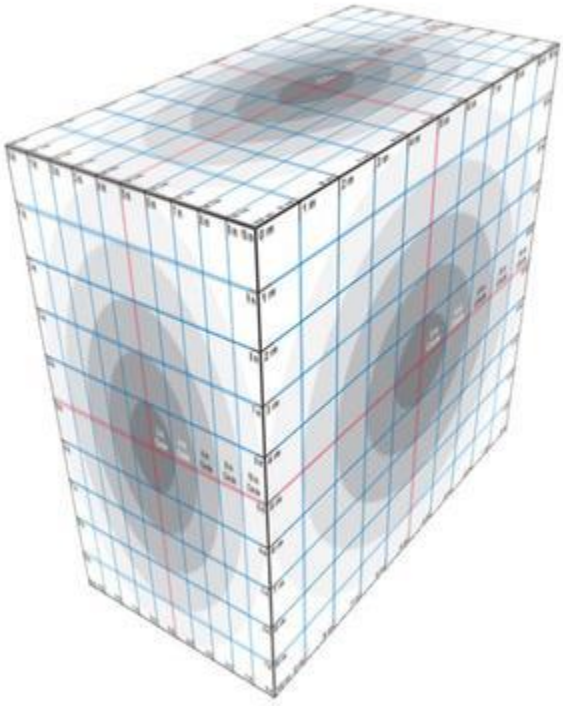
Now that you've begun texturing, you might be wondering how to make those textures look just right. How do you make each side (or face) of an object look proportional. That's what this section is all about.

Instruction
When you apply a texture to a prim, it will completely fill each face exactly one time. If all the faces of your prim are the same size, it's no problem - the texture looks proportional (the same size and "stretch") on all sides.



But what if you make something like a rectangle with different sized faces?

In this example, the cube from above was halved at the Y axis, making it half as wide as the other. See how the textures on the top and side now look "squished"? The texture is no longer proportional.

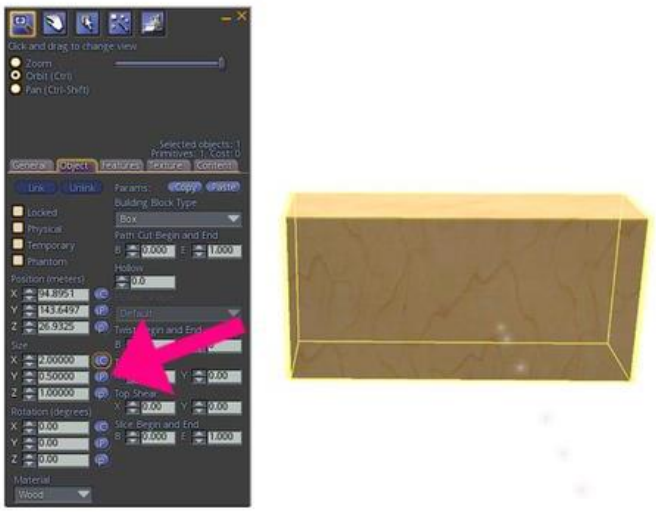


You can fix that problem by adjusting the number of times a texture repeats on each side of your prim. It's easy - and it will make your creations look so much better!

Practice

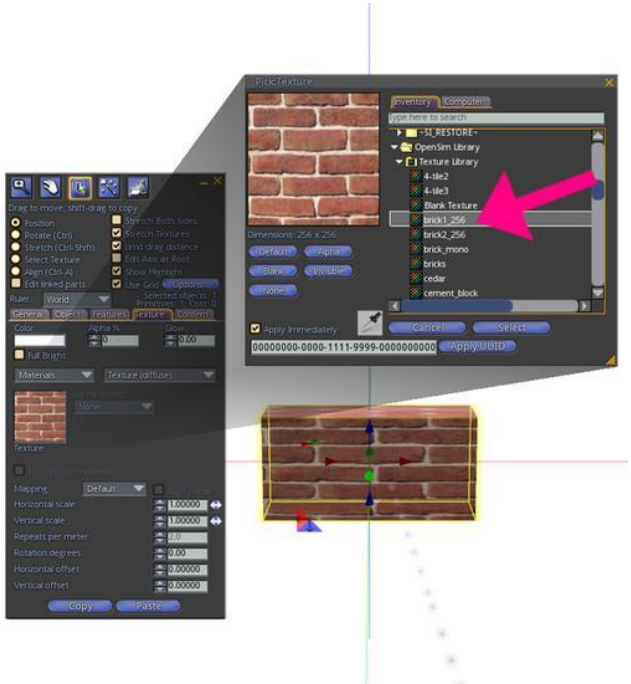
Step 1: Creating a prim with different sized faces
When you create a prim with different sized faces, you will notice that the texture on each face "looks" different.

Rez a plywood cube, and in the Object tab of the Edit window, change the size parameters to X=2, Y=.5, Z=1.

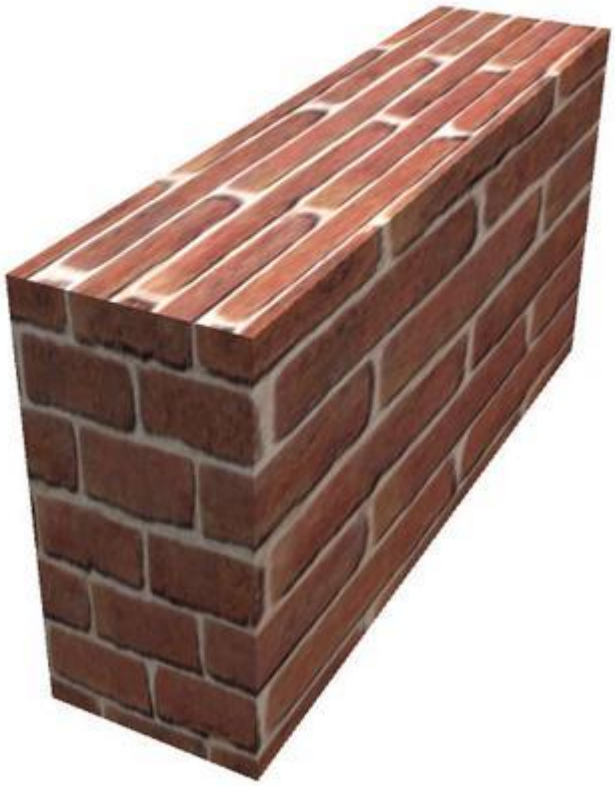


Step 2: Applying a texture to all faces
Depending on the texture you choose, you may notice a big difference in how a texture looks on each face.

Let's apply a texture to our prim. Right click to edit the prim, and in the Edit window, go to the Object tab > Texture box > click on the box to open the Texture Picker Window. Then, in your inventory, go to the OpenSim Library folder > Texture Library and choose the "brick1_256" texture.



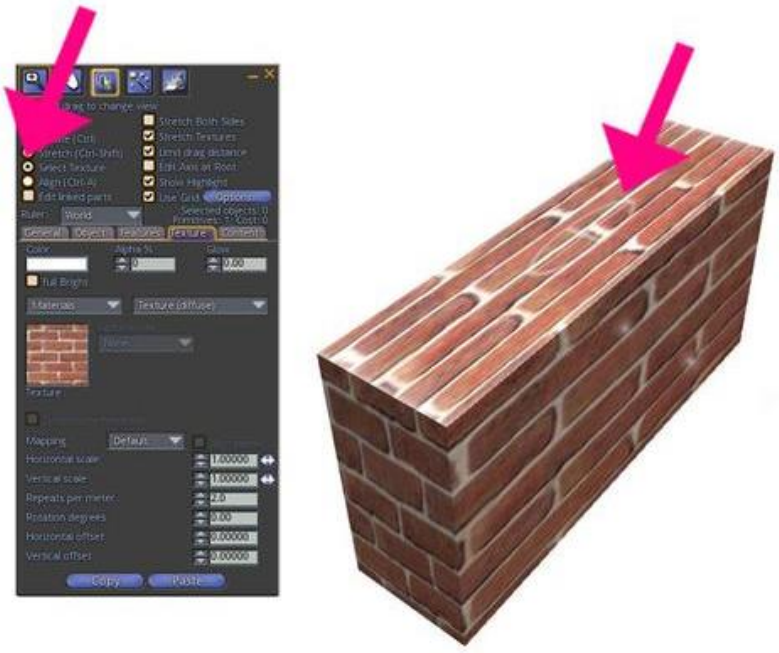
Look at each face. Do you see how the bricks on the ends of the wall look squished? On the shortest side, the bricks look like they have been squished in, making them look short and stubby. On the top, the bricks look like they have been stretched out, making them look long and narrow.



Step 3: Adjusting the texture repeat

To make a texture look right (or proportional) on each face of your wall, you can change the repeat. In the Texture tab of the Edit window, the repeat is usually set to 1.000, which means the texture will repeat exactly one time. You may use the arrows to make the texture repeat fewer or more times.

With your wall in Edit mode, make sure the Select Texture radio button is selected so we can work with just one texture on one face of the prim at a time. Click on the top of the wall where the texture looks stretched to make the bricks look narrow - you should see a faint white target symbol appear ONLY on the top of the wall.



Now find the Horizontal and Vertical repeat (or scale, depending upon your viewer) settings in the Texture tab, and change the Horizontal scale to 2.0 and the Vertical scale to .5. Do the bricks look more proportional now?

In order to fix every side of the prim, you would need to change the repeat or scale for each face of the prim. If the default scale of a texture is 1.0, but the prim is only .5 meters wide, then what scale should you use for the sides of the wall to make the texture proportional?

See if you can figure out the right answer, but if you need a hint, scroll to the bottom of the page!

Step 4: Flipping a texture

Sometimes, you may get a better look by completely flipping a texture. You can do this by checking the Flip box next to the Repeats Per Face settings.

Select just one face of your wall. Check and uncheck the Flip box for the horizontal and vertical repeat settings.

Rez a few different shaped prims and apply different textures to them. Practice changing the horizontal and vertical repeats. Do you find you like oversize, fantasy-like textures or more realistic ones?

Answer Hint: The correct answer is .5! You want the texture scale to match the size of your object if the proportion is 1:1. But let's say you wanted the bricks to appear SMALLER than the default size of the texture. In that case, you would want to shrink the texture scale proportionally to the prim's size across all the faces of the prim.

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Offsets

Someone once said, "Variety is the spice of life." If everything looks like it was stamped out of the same mold, it doesn't look real.

Instruction

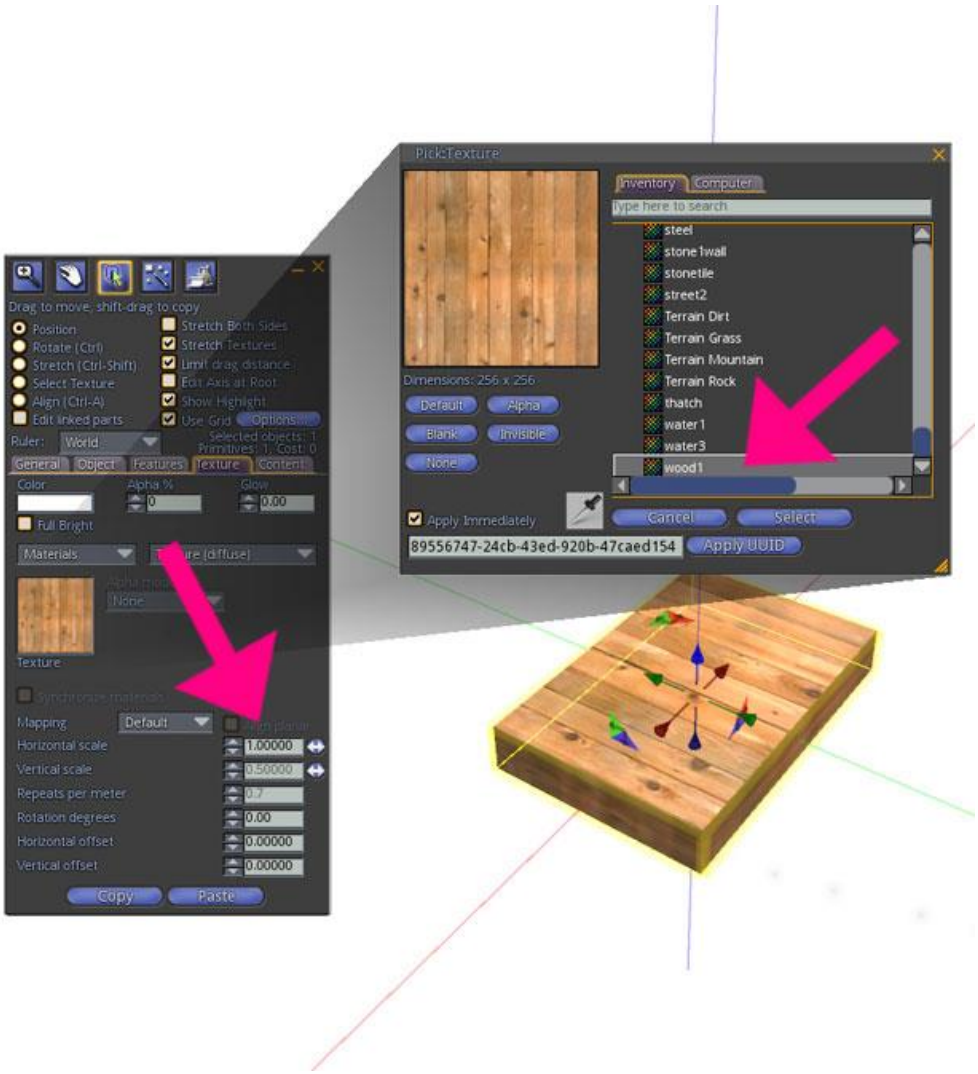
You can add variety to your texturing by using the Offset feature in the Texture tab. Offset will allow you to slightly change the placement of the texture as applied to your objects.

Practice

Let's say you want to make some wood steps. In real life, you know they would each have the same wood pattern. But you also know that they would look different from each other.

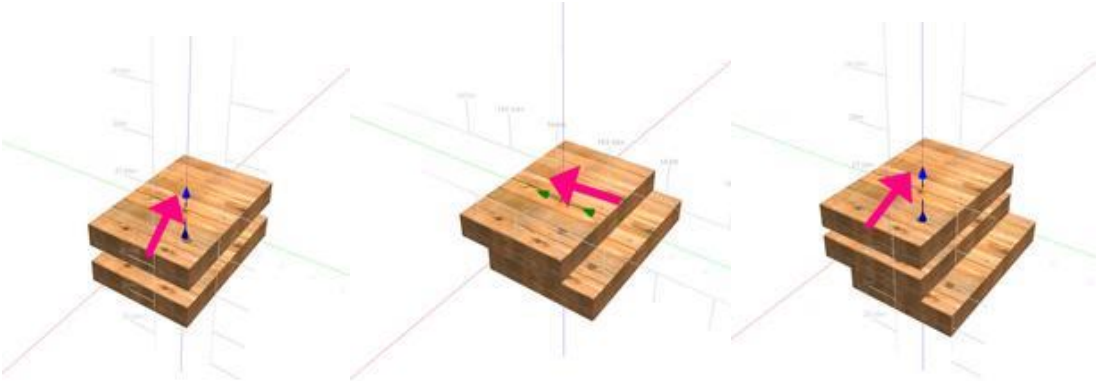
Step 1: Making some steps
Rez a cube. Stretch and flatten it to form a step. The step in our example is:
X: 1.500
Y: 1.000
Z: 0.225

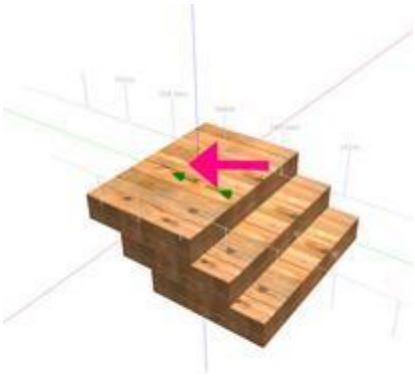
Step 2: Texture with a wood pattern
In the Edit window > Textures tab, click the Texture box to open the Texture Picker. In the OpenSim Library > Texture Library, select the texture named "wood1" and adjust the front repeat of the step to:
Horizontal: 1.000
Vertical: 0.500



Step 3: Duplicate Prim and Move to Create Stairs

Now hold down the shift key and drag up to duplicate the step. Duplicate and position three or four steps to form stairs.

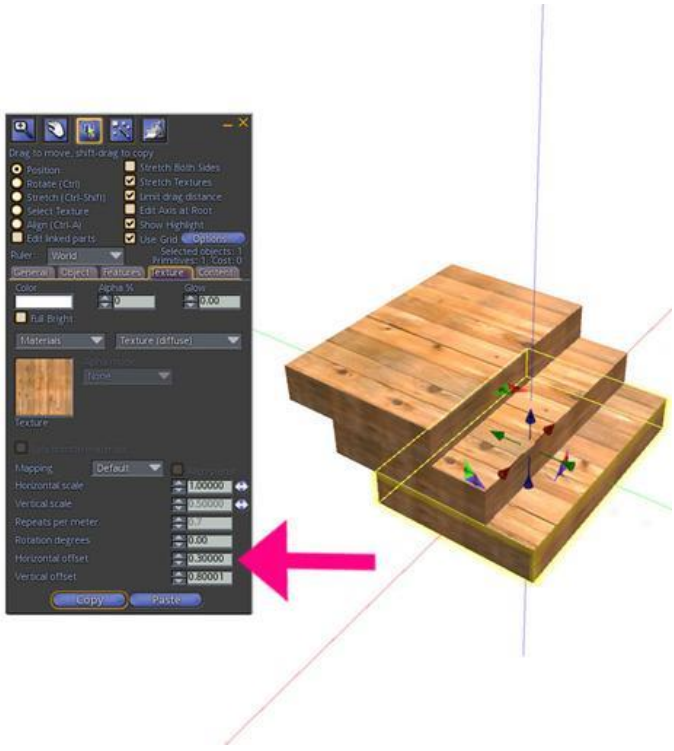




NOW: Do you see how each step looks exactly the same? It looks pretty good - but not great.

Step 4: Using Offset
You can use Offset to adjust the texturing on an entire prim, or just one side at a time. It will depend on the texture and look you want. In real life, these wood steps would look like they were each made of different boards instead of the same ones. Let's change how our steps look to make them look more natural and life-like.

In Edit mode, select the bottom step. In the Texture tab, change the Offset to:
Horizontal: 0.300
Vertical: 0.800



Select the other steps one at a time. Change the Offset differently for each one. You don't have to use "round" numbers. When you're done, you'll see each step looks slightly different than the others, lending a touch of realism to our steps.

Tip: Combining Offset with Flip might give you even better results!

You now have the power to modify, or change, the texture of objects!

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Fixing Flicker (no, not the photo sharing service)

Great texture artists are picky about the details. Have you ever noticed something in the virtual world that looks like it is flickering strangely or a dotted line appears between two adjoining prims? Distracting, isn't it?

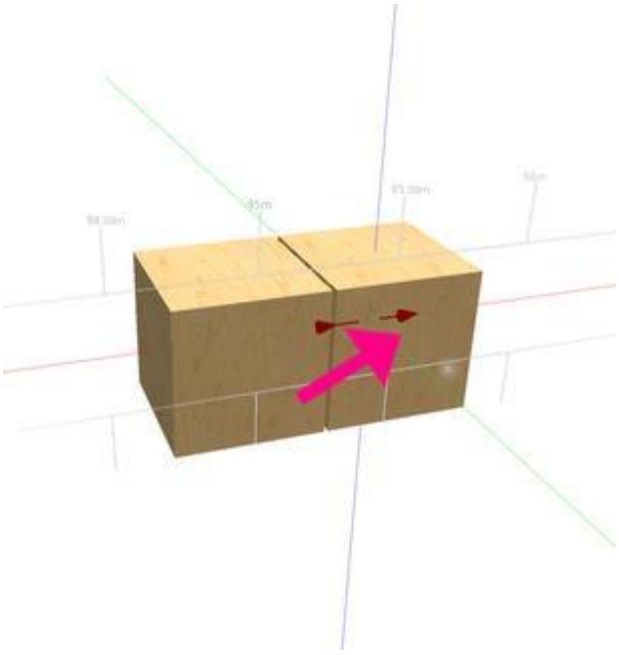
Instruction

Texture flicker happens when two adjacent textures are competing visually on your computer screen. Depending on the angle from which you are viewing your object, one prim or another will seem to win. It's very easy to fix, and one mark of careful texturing!

Practice

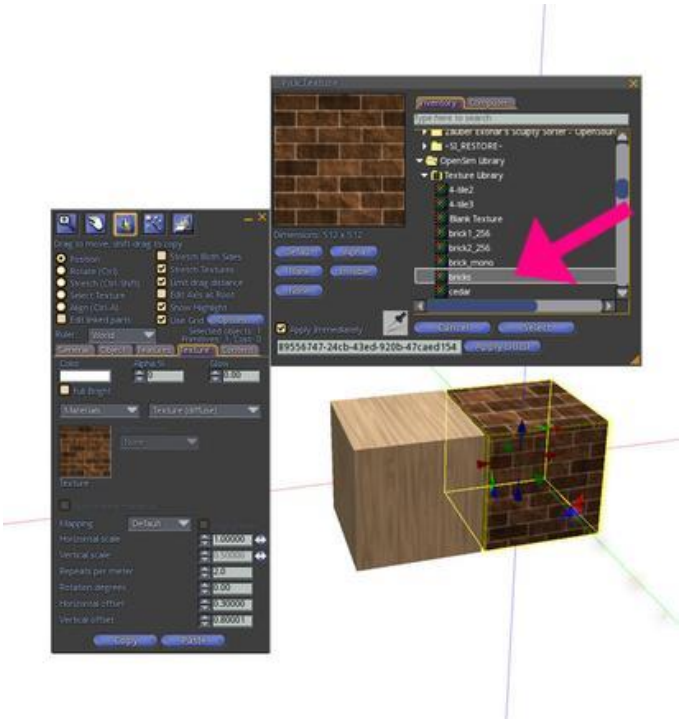
Step 1. Create two adjoining prims

Rez a cube, then hold down the shift key a drag along the X axis to duplicate that prim. Position the duplicate so it's touching the first prim, as if you are building a wall, one prim at a time.



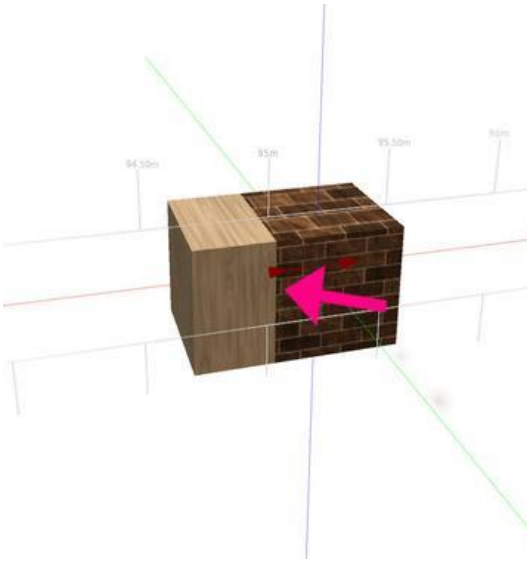
Step 2. Texture the Prims with Different Textures

Now, one at a time, select the Texture box in the Edit Window > Texture tab and select a different texture from the OpenSim Library > Texture Library for each prim. In the example, the first prim is textured with "cedar" and the second prim is textured with "bricks".



Step 3. Drag the Second Prim "Half Way" Into the First Prim

Now, to show the flicker effect, drag the second prim about half way INTO the first prim, so that some portion of both prims are overlapping each other (occupying the same virtual space).



You probably saw the flicker effect happen as you were dragging the second prim into the first, but even after you let go, as you walk around your prims, you will see the flicker happening again as your computer tries to decide which texture to show from any given angle. This happens because parts of the two prims are occupying exactly the same virtual space at the same time. So which texture should be on top? Your computer can't tell, so it "flickers" between the two textures depending upon your viewing angle.

Step 4. Fixing the flicker
The only way to remove the flicker effect is to make sure that no two prims occupy the same virtual space at the same time. For objects that are right next to each other, you need to move the ends so that they are just touching - but not overlapping - each other. Sometimes this can be kind of annoying to fix by using the arrows by hand, but using the Position parameters to change the numbers can help.

Another option to get rid of flicker is to ever so slightly adjust one of the sections along a different axis, so one prim is just slightly "in front of" or "behind" another prim. It's best to do this with the Position Parameters so you can change the position with high accuracy. You can change the measurement by as little as 0.002. It won't really change how your build looks - but it will get rid of that annoying flicker!

Change the alignment of one of the prims you created just slightly. Look at it from different angles to see if you solved the flicker problem.

You may have noticed flicker on some of your early creations. If you have time, go back and use your new knowledge to fix those flicker problems.

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Advanced Texturing Techniques: Beyond the Basics

Overview
By now you have the basics of texturing down, but there are a few things you can do to really take your texturing to the next level.

Learning Objectives
By the end of this module, you will have the following skills:

- Making an object look shiny or bumpy
- Adding shadows and light effects to objects

You will demonstrate your new skills by:

- Making objects that look shiny and bumpy
- Making a shadow for an object
- Making an object that glows in the dark

The added tips in this section will let you add new textures to your collection and use them with magnificent effect!

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Shiny and Bumpy

How can you make a castle wall look rugged or a gleaming statue look shiny?

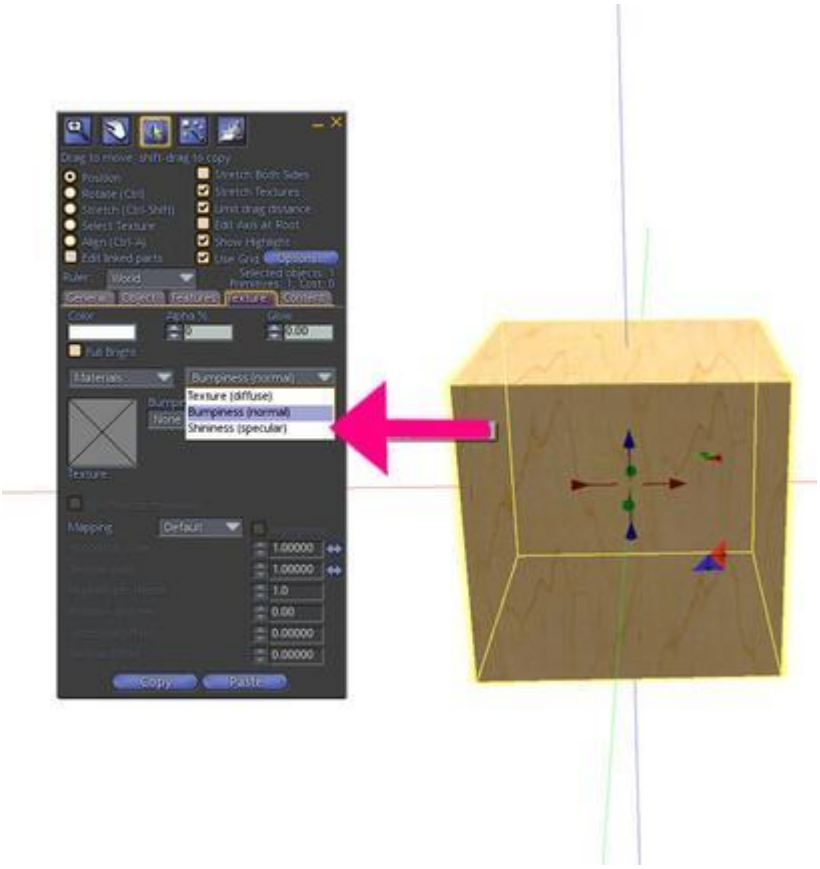
Instruction

In addition to choosing just the right texture for the object you are building, you can also control how Shiny and Bumpy it appears. Adding these features to your object will add to its realism and make it really pop! (However, the ability to see these features will depend on the viewing preferences of each individual viewer. Computers with less powerful graphics cards may not have Shininess and Bumpiness active.)

Practice

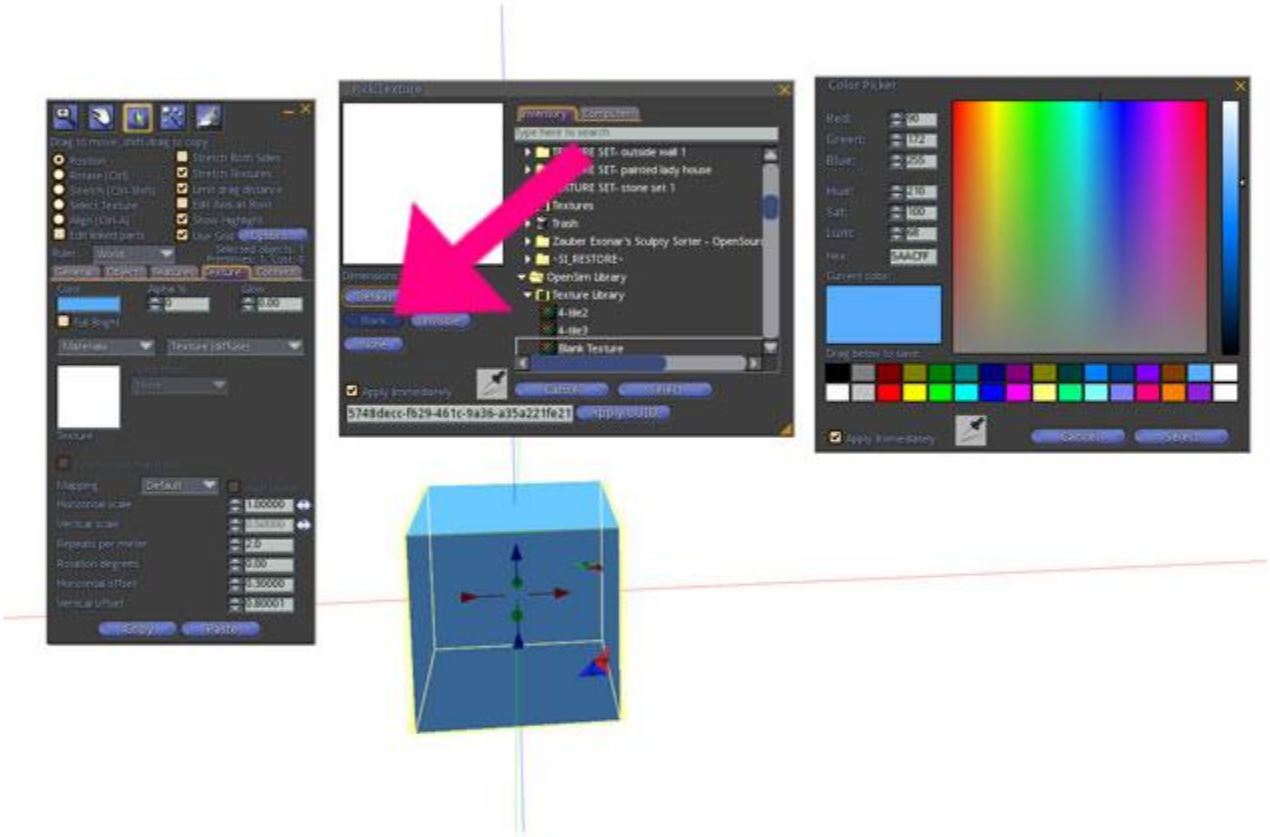
Step 1. Shininess and bumpiness

The drop down menus for Shininess and Bumpiness are just below the Texture and Color pickers in the Texture tab in most viewers. You may also see a drop down menu for Mapping or Materials. The placement and label of these settings vary from viewer to viewer, but most viewers will have a few options.

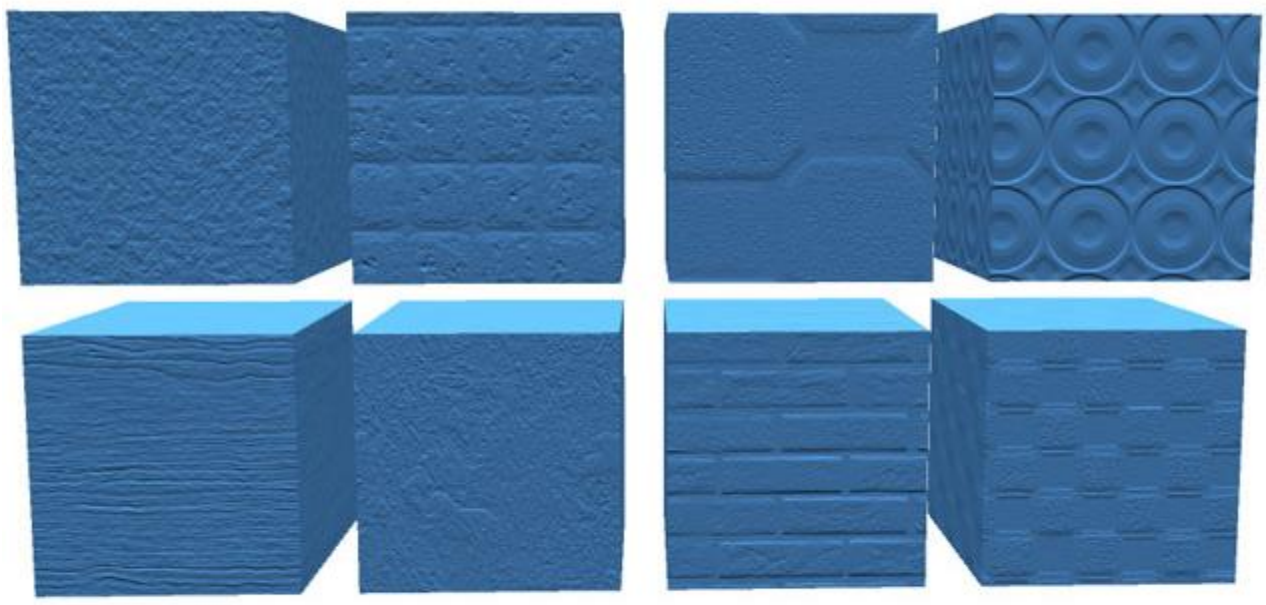


Step 2. Bumpiness
The Bumpiness drop down menu offers several bumpy textures to use in addition to the texture you choose.

Rez a prim. It will probably be easiest to see the difference between the different bumpy choices if you texture the prim with a blank texture (plain white) and then give it a bright color.



Now duplicate it a few times. Set each prim to a different bumpy texture and compare.



Each texture looks quite different with different bumpy textures applied!

Step 2. Shininess
The Shininess drop down menu offers levels of shininess: None, Low, Medium, and High.

Rez a prim. Texture it and duplicate it. Set one of the prims to Shininess and compare.

The plain purple prims have different shiny settings, from none on the left to high shiny on the right.

Can any of your early creations benefit from a shiny or bumpy touch? Go add them now!

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Light and Shadow

In real life, objects have shadows and can be lit up from the sun - or from within! You can add a layer of depth to your objects by using shadows and light in amazing ways.

Instruction

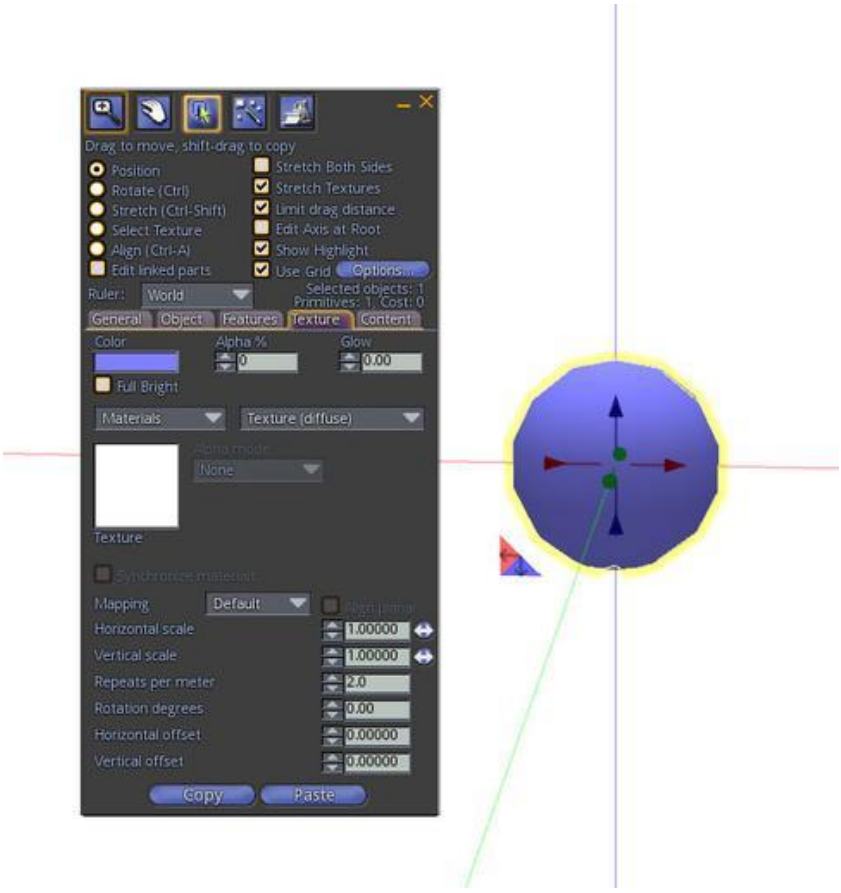
You have already learned how to tint faces of prims to show shadows and light. The inside of a cut prim might be tinted darker. The top face of a step might be lighter. Although there are some sophisticated ways to add shadows and light, you can easily add a basic shadow, make it bright like it's emitting light, or even make something glow.

Practice

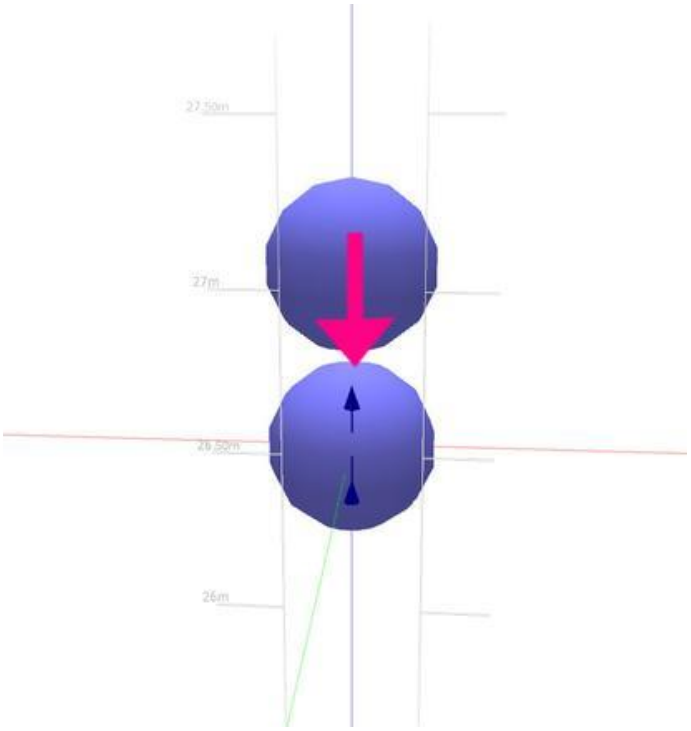
Step 1. Add a basic shadow

To add a basic shadow, you can make a prim the same shape as your object, flatten it, tint it and position it under your object.

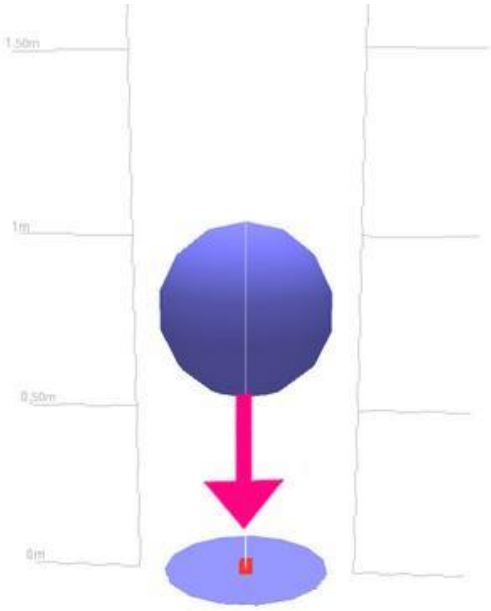
For this example, rez a sphere, give it a texture, and move it up in the air a bit so it isn't touching the ground.



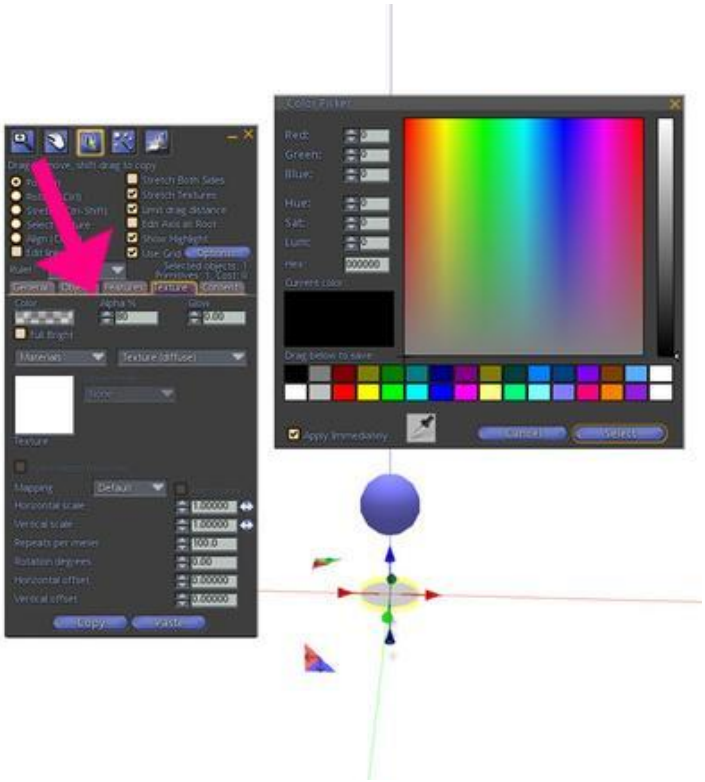
Now duplicate the sphere by holding down the shift key and dragging down on the Z axis (blue arrow) so that you have a new sphere below the first.



Edit the bottom sphere, and hold down the CTRL+SHIFT keys to "squish" the prim by dragging down on the X axis (red) handle to flatten the sphere.



Now, change the color or tint of the bottom, squished sphere to make it black or a very dark grey. Then set the transparency to 80%.

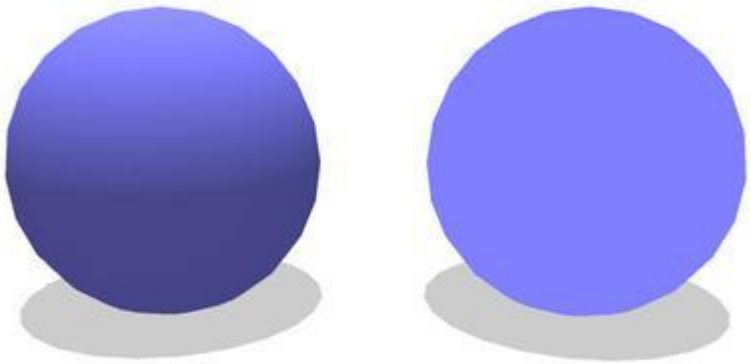


Finally, position the top sphere just on top of the bottom squished sphere. It looks like the top sphere is casting a shadow!

Adding shadows to items can dramatically increase the level of realism in your builds and scenes!

Step 2. Adding Full Bright
More sophisticated lighting features are covered in the building modules; however, you can make a prim light up using the Full Bright feature in the Texture tab. This is especially useful if you have an object you want people to see no matter what time of day or night they view it.

Duplicate the sphere and shadow we created above (hold down the SHIFT key, select both prims, then drag along the X axis (red arrow) . Now, select the second prim and check the Full Bright feature in the Texture tab. That's all there is to it!



Use Full Bright cautiously! Novice builders often make the mistake of overusing the Full Bright setting, which can make night scenes look garish and oversaturated. Full Bright is best used on signs, light sources, video screens, and other objects that might naturally emit light.

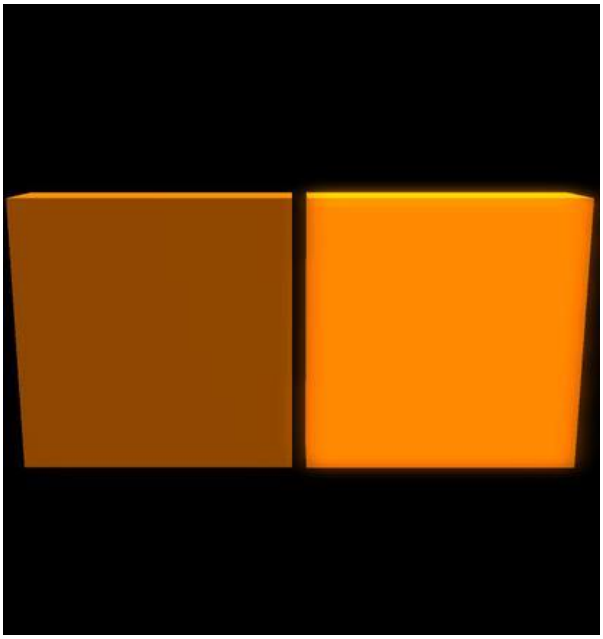
Make sure to always view your builds in both daytime and nighttime settings so you can see the difference. A realistic night scene has dark shadows, so be careful about drowning the shadows out with Full Bright!

How can your creations benefit from shadows or Full Bright? Take a minute to add those features now.

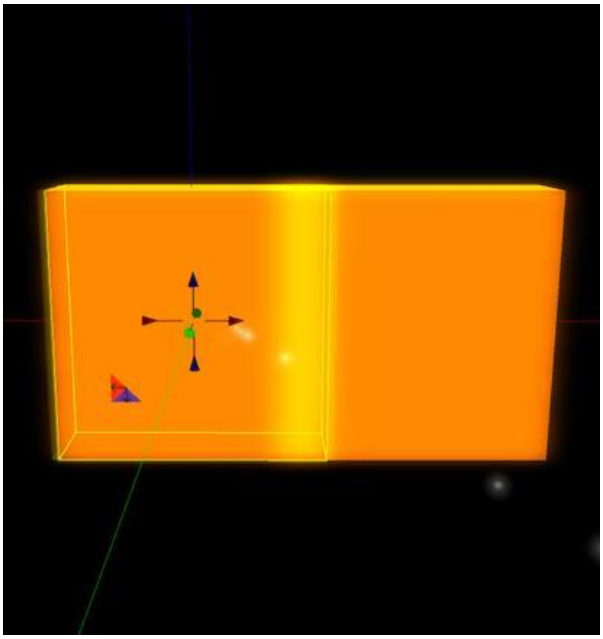
Step 3. Playing with Glow

In addition to making the object Full Bright, you can even add a glow to your object to further enhance the lighting effect.

Rez two prims, make the textures blank, and color them orange. Leave the Full Bright feature UNCHECKED. Now increase the Glow parameter to .10 on the prim on the right. Can you see the difference?

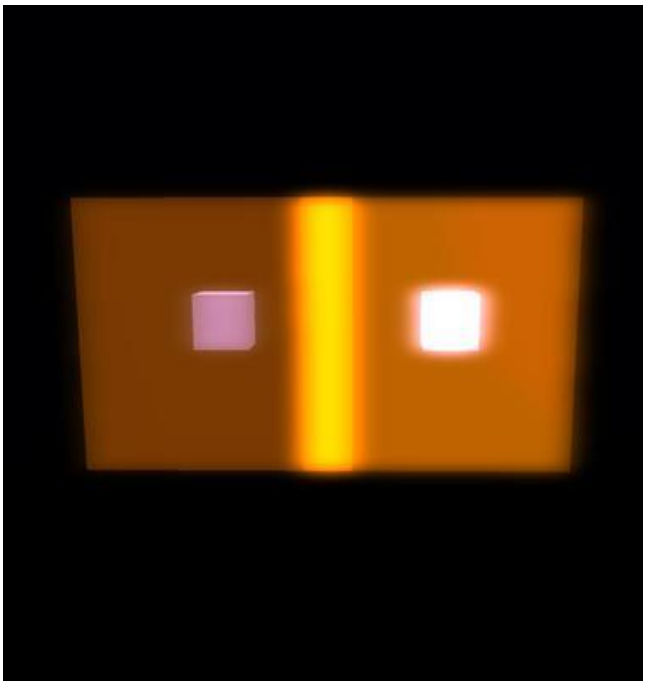


Now add .10 glow to the object on the left, and drag the prims so they are slightly overlapping each other. The glow gets much more intense where the objects intersect!



Glow can be especially fun to play with on prims that are semi-transparent, too! In the example below, two colored prims sit behind the two glowing panels. The glow panel on the right has a higher level of glow applied, and the panel on the right has a higher level of transparency applied.

Front view:



Overhead view:

Use Glow cautiously! Another mistake novice builders often make is overusing the Glow setting, which,

like Full Bright, can make both daytime and night scenes look garish and oversaturated. Glow can even ruin the view of other parts of your build! You can do very interesting things with glow, particularly by layering different prims using combinations of transparency and glow, to make neat and fantastic builds, but again like Full Bright, Glow is best used cautiously or on signs, light sources, video screens, and other objects that might naturally emit light. Make sure to always view your builds in both daytime and nighttime settings so you can see the difference.

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Using Textures Wisely and Well

If you put into practice everything you've learned so far about texturing, you will be known for making excellent objects. However, there are just a couple more things you should know to use textures like a pro.

Using Too Many Textures or Lots of High Resolution Images are the #1 source of Scene Lag!

The number and size of textures you use to texture an object or scene can affect the amount of *lag* someone experiences as they walk through your build. Lag is a slow down in performance of the viewer. If you've ever tried to move or walk and you find that your avatar isn't responding as quickly as usual, or that everything just seems sluggish, or the objects in the scene look grey and take a long time to come into focus, that might be because the builder wasn't careful and used either too many textures, or textures that were too high in resolution!

Textures that haven't loaded or "rezzed" yet look grey.

A simple rule to remember is the more textures you use, the more lag visitors will experience. Using as few textures as possible, and textures that are as small as possible, can make everything come into focus much more quickly and make the experience much more pleasant for your visitors.

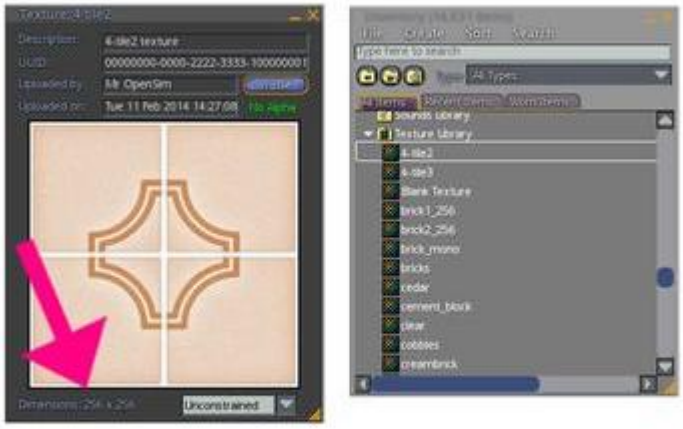
Practice

An image's size, or resolution, refers to how many pixel columns and rows appear in the image. The higher the resolution, the more pixels there are in the image, and the bigger the image's file size becomes. As with anything else you download on the internet, the bigger the file size, the longer it takes to download, and the same thing happens with your viewer! If a builder uses a lot of high resolution images, or uses a lot of different images, then the viewer takes more time to download and display the images for the user.

For the mathematically inclined, viewers display images in factors of 16, so no matter what size or resolution the image is on your hard drive, when you upload the texture to the OpenSimulator server, it will convert the image to the closest factor of 16.

Step 1. Finding the size (or resolution) of a texture
When you double-click on any texture in your Inventory, a Texture Preview window will open. At the bottom left is the size (resolution) of the texture. The larger the numbers, the greater the size (resolution) of the texture.

Open your Inventory and navigate to the OpenSim Library > Texture Library folder. Now find the texture called "4-tile2" and double click to view it.



See the Dimensions at the bottom? This texture is 256 pixel columns across by 256 pixel rows horizontally.

Tip: Choosing smaller textures for most of your build will make it come into focus faster and lessen lag. Save the largest textures for when you need a lot of detail. For most projects, you can use textures that are 256 x 256 or even 128 x 128.

Professional builders pay very close attention to the size of their textures and use the smallest textures possible for things like walls, trim, floors, and other background or general scene objects. They save larger textures, like 512x512, for objects that are areas of focus or items that need to show more detail. The largest texture size you can use in most OpenSimulator grids is 1024x1024, and that should be reserved for signs with text, large maps, or other items that really do require a higher resolution to view properly.

Note: If you are handy with Photoshop or another image editor, resize your images before you import them! Using a texture that is 256 x 256 will be faster than using one that is 512 x 512.

Step 2. Looking at the number of textures in a build
Let's say you're going to build a house. How many textures will you need to use? Perhaps you will need a siding texture for the outside, a wooden floor texture, and a wallpaper texture for the inside.

Can you re-use the wooden floor texture on some of the furniture inside too? And maybe use color tinting and offset to make the same texture look a little different?

If you think before you texture, you can reduce the number of textures you use in a build. Ask yourself:

- Can I use repeat, offset, rotation or tint to make the different faces look unique?
- What about using shininess or bumpiness to vary a look?
- Is there a specific part of a texture I can use to make a prim look different from other prims in my object?

Keep in mind that when you upload a texture, the server assigns that texture a unique identifier, known as a UUID. If you upload the exact same texture a second time, the second upload will be given a new UUID. If you use both textures in your build, the server will have to send both files to your viewer, so it will take twice as long to display even though it is the exact same image! This is why it is important to think carefully about how many textures you use!

Tip: If you are doing a big build as a group, decide together on the textures you will use, and be sure to only upload the image once and then share the same texture file among everyone on your build team. Using fewer textures will really make a large build appear f-a-s-t!

Using Poorly Made or Garish Textures Can Make a Great Build Look Horrible!!

In the real world, someone could build a perfectly nice house, but if the person who comes to paint it isn't a professional and does a poor job, then that beautiful house can be the ugliest eyesore in the neighborhood. No one wants to live next to a neighbor with a garishly painted house, either!

The same is true of your builds in the virtual world! Nothing can ruin a great build faster than poorly made or garish textures.

If you are going to be building in a community or with other people, then you want to make sure that your textures or aesthetic style isn't going to clash with everyone else's tastes.

If the whole area uses very realistic textures, and your textures look cartoony or abstract, then visually, your build doesn't blend in well with the community and that can be very disruptive for visitors.

Make sure to talk with your neighbors and be observant of the style of architecture and textures they use. You can still be creative and unique, but don't be the neighbor with the crazy weird house!

Attention to Detail is the Mark of a Great Builder!

When you get busy building, it's easy to get lost in the big picture of putting all the elements of a scene together, but if you fail to pay attention to the little details, all those little shortcuts - sloppy corners, bad texture alignment, flickering overlapping textures - will turn your great build into something that's only mediocre.

Look at this walkway path intersection. The texture the builder used for the path is ok, but the builder didn't take the time to make sure the texture repeats were properly set, so the bricks on some parts of the path appear to be squished or stretched. This really distracts from the experience of walking on the path - it's not only a little disorienting, it also doesn't look very nice.

Being a great builder takes time, care, and patience, just like being a carpenter in real life. You can't rush great quality, it only comes from taking the care and time to make sure that prim adjoins the next prim just so, and the texture on one face lines up just right with the texture on the other face. We all take a shortcuts now and then, but take too many, and it really degrades the quality of your work.

Look at these two kitchen models. It's the same, identical build, they both are textured nicely, but the second one looks a little nicer, a little more professionally done. Can you tell the difference?

The kitchen on the bottom used a slightly shaded texture as the base for all the cabinet faces, and it makes it seem as if light is reflecting off the surfaces, giving the kitchen a little more depth and variation. These simple tricks, and paying attention to the details of your textures, can move your builds from good to great!

You now know more than most people do about texturing! Good job! Just practice what you've learned and you will create objects that people love to look at.

If you are using the **PRIMLAND** Tutorial game, stop here and continue on the path!

Scripting

The OpenSimulator Scripting Language (OSSL) and Linden Scripting Language (LSL) allows you to program interactivity into your virtual objects. Without this programming, objects are motionless and non-interactive. To make an object interactive, simply add a script. Using OSSL or LSL, you can create cars, planes, amusement park rides, weapons, and other entertainment devices.

Here are some good resources for beginners:

LSL wiki - http://wiki.secondlife.com/wiki/LSL_Tutorial

Video Tutorials - http://wiki.secondlife.com/wiki/Video_Tutorials

OSSL Functions - http://opensimulator.org/wiki/Category:OSSL_Functions

Important Concepts

Before you start scripting, you need to understand a few important concepts and terminologies.

Inventory and Scripts

Each prim can store other things inside itself. You can think of the prim's inventory as its pocket or closet. In its inventory, a prim can store other prims, objects, notecards, and scripts. In OpenSimulator, a script inside a prim can control the prim's position and orientation, color, shape, and size, and other physical properties. It can enable the prim's response to commands, reaction to the environment, or any other interactivity provided by the scripting language.

Physics and Vehicles

An object has a number of properties, including material. Material is especially important when you are scripting an object that behaves like a physical object, for example, a car. Any moving object that has physical properties is called a vehicle. Please note that as of the time of this writing, OpenSimulator offers two different physics engines for grid operators, the Open Dynamic Engine (ODE) and BulletSim. Which engine your grid is using depends on which engine the grid owner chose when they set up the grid. You may need to ask your grid owner. For more information about physics in OpenSimulator, see the physics wiki page.

States and Events

In OSSL and LSL, most scripts sit idle until they receive some input, or detect changes in their environments. At any moment, the script is in some state, and will react to events or inputs according to some scheme defined by the programmer. However, a script can also contain two or more different states, and react differently to events or inputs depending on its particular state. One common abstract model that is used in such cases is called a Finite State Machine.

For example, a door might be in a waiting state, and ignore all inputs except being touched. Once touched, it goes to the open state, in which it ignores being touched, but monitors which avatars pass through it. After a while, it changes to the closing state during which it closes, and then returns to the waiting state. States are not the only way to represent this kind of behavior, but in some cases they are a very good way.

In OSSL and LSL, a state is a specified section of code within which all events are specified. The main

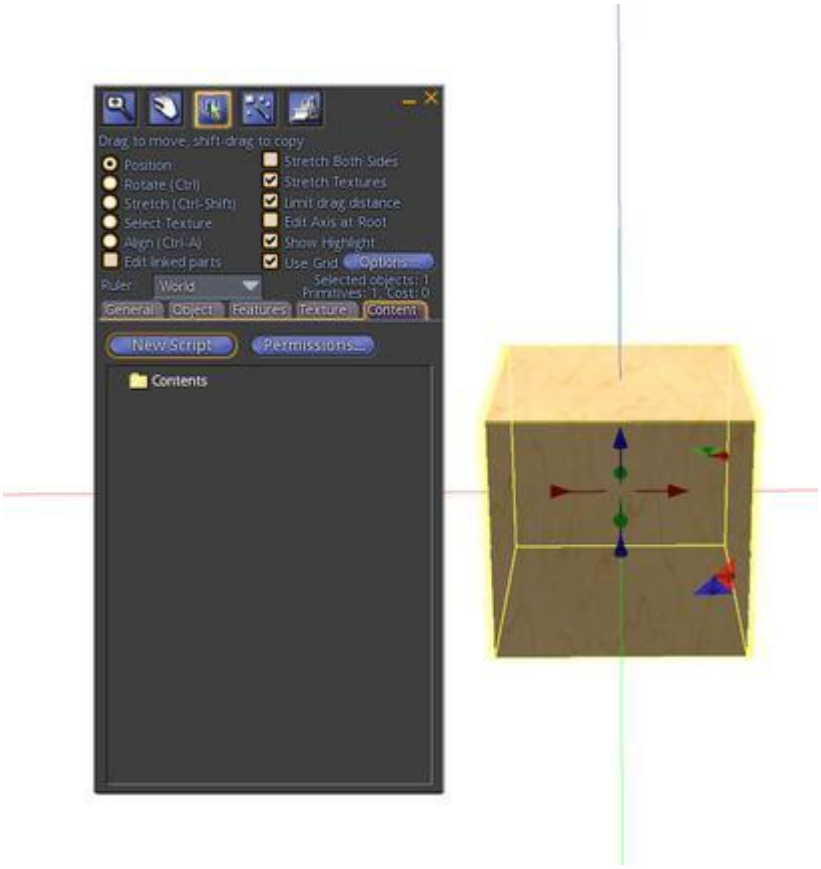
state that is required by all OSSL and LSL scripts is called default. All scripts must have a default state, and every state must have at least one event.

Here is a link to more detailed information on the "State Machine" that is LSL:

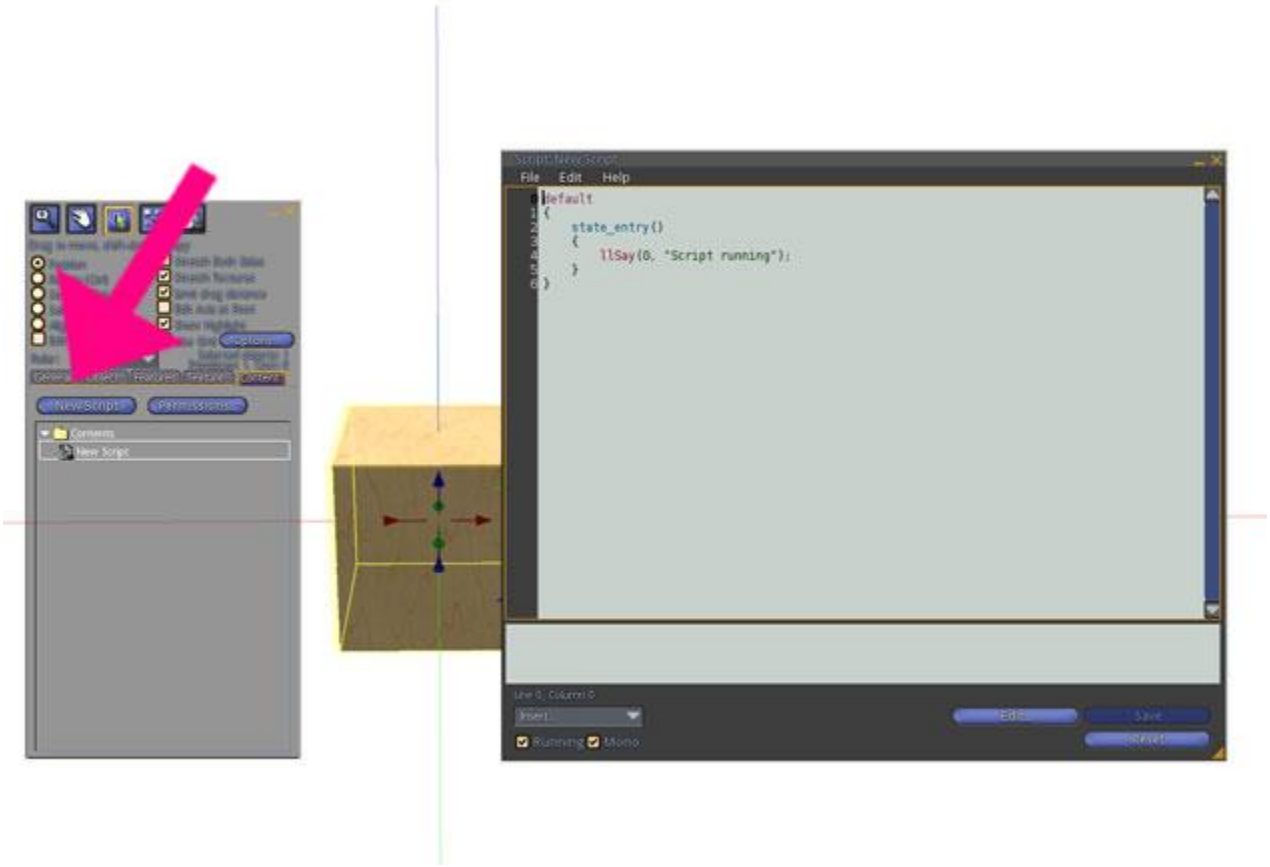
http://lslwiki.net/lslwiki/wakka.php?wakka=state

Running OSSL and LSL Scripts

OpenSimulator includes a facility for defining scripts within existing objects. To begin with your first script, rez a prim and in the Edit window, click the Contents tab. This directory is where scripts, if any are associated with an object, will appear.



Left-click on the New Script button, and a Script editing window will open. The editing window has a script editing area, where you can type or paste in a new script, and buttons for saving the script, undoing changes, etc.



The script editing area should contain the following default script:

```
default
{
    state_entry()
    {
        llSay(0, "Script running");
    }
}
```

Left-Click on the Save button. The message "Compile successful, saving..." should appear in the box below the script editing area. After a short pause you should see "Save complete" appear in the box.

Once the script is saved, you will see the message "Script running" appear in the text chat, usually in the lower-left-hand corner of your screen. This message indicates that the script has started.

Entering and Running a Simple Script

Here is a very simple program that changes the color and size of the object every time the object is touched.

```
integer counter;

default
{
    state_entry()
    {
        llSay(0, "Script running");

    }



    touch_start(integer total_number)
    {   // do these instructions when the object is touched.
        counter = 0;
        counter = counter + 1;

        // choose three random RGB color components between 0. and 1.0.
        float redness = llFrand( 1.0 );
        float greenness = llFrand( 1.0 );
        float blueness = llFrand( 1.0 );

        // combine color components into a vector and use that vector
        // to set object color.
        vector prim_color = < redness, greenness, blueness >;
        llSetColor( prim_color, ALL_SIDES );   // set object color to new
color.

        // choose a random number between 0. and 10. for use as a scale
factor.
        float new_scale = llFrand(10.0) + 1.0;
        llSetScale(< new_scale, new_scale, new_scale > ); // set object
scale.
        llSay( 0, "Touched by angel number " + (string)counter);
    }
}
```

As you click on the box, it will change with different colors and sizes and this program will count the number of times the object is touched, and write the number in the lower-left-hand corner of the screen. To insert this script into the object you just created, right-click on the object, reopen the Script editing window, paste this script into the Script editing area, and click Save. When the script has been saved, test it by right-clicking on the object and choosing Touch, as before.

You may have to do some debugging to correct errors, and when you finally get things running correctly, you may want to right-click on the object and choose Take to move the object into your Inventory. Objects in your inventory will remain there even after you log off, and will be available when you log back in.

When a touch_start event occurs, the program starts a timer that runs out every two seconds, causing a timer event. Whenever a timer event occurs, the script segment within the brackets below the string "timer()" is executed. The timer() section counts the number of times it is activated, and resets the script after twenty activations.

If multiple avatars touch the object at the same time, strange behavior may occur. Such an event may be

monitored and controlled by using the total_number variable passed to touch_start.

Summary

This is just a very, very simple introduction to what can be accomplished in Scripting. Scripting and programming is a much broader topic than this manual can cover, so we encourage you to explore the many resources on the web to learn more about scripting. There are many varieties of scripts there that can be reverse engineered to fit your specific needs.

To learn more about scripting, see these links:

LSL wiki - http://wiki.secondlife.com/wiki/LSL_Tutorial

Video Tutorials - http://wiki.secondlife.com/wiki/Video_Tutorials

OSSL Functions - http://opensimulator.org/wiki/Category:OSSL_Functions