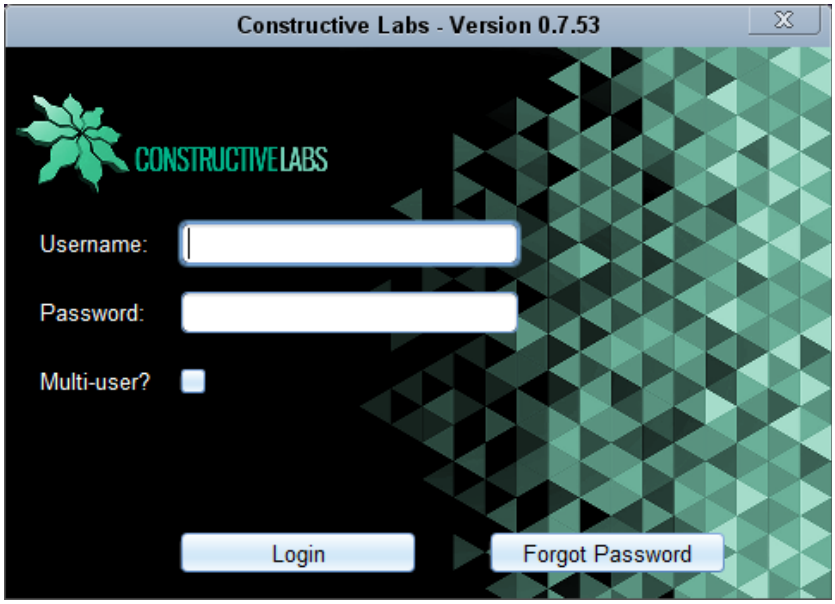
How to Use Constructive Labs Monkey

This doc is way outdated, please use the current documentation at http://constructivelabs.com/wiki

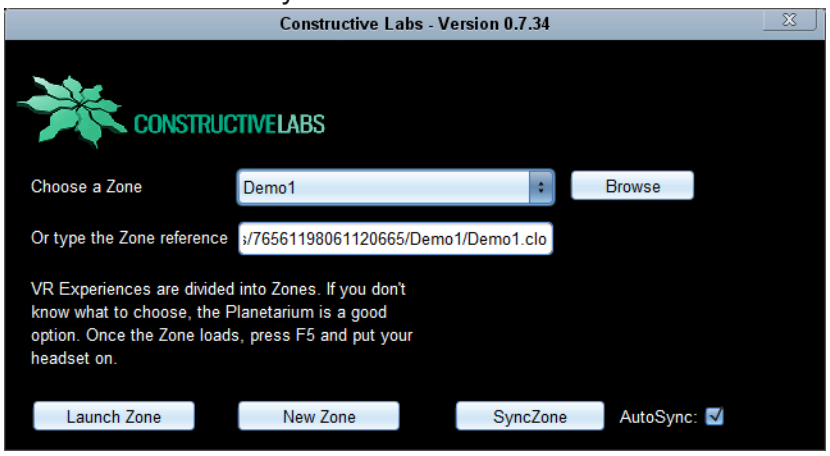
Table of Contents

Getting Started

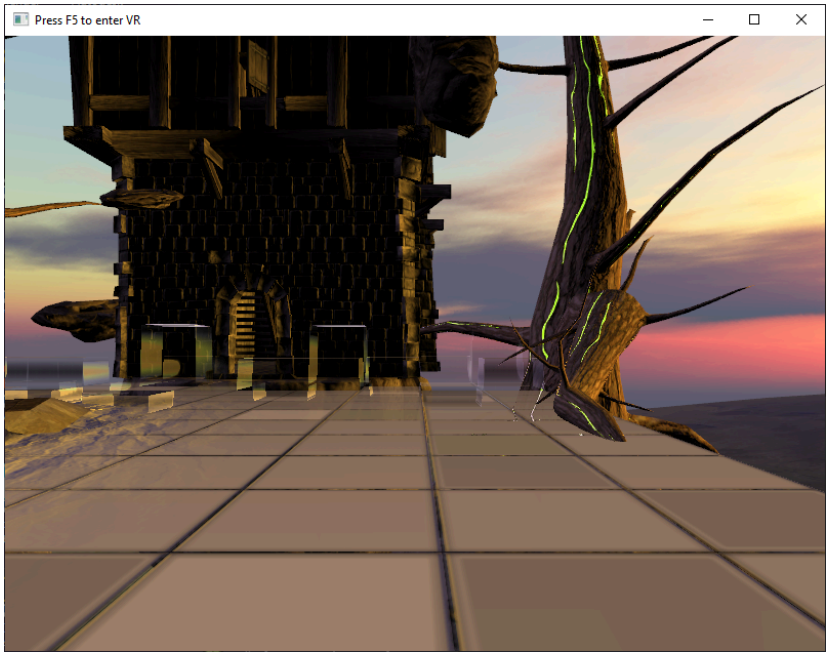
Login with your portal.constructivelabs.com account



Choose a Zone with your mouse to start.



Click the Launch Zone button. The zone will load and you will see it displayed in a window:



You can move around the scene in the window with your keyboard arrow keys or the WASD keys. The right-mouse button (drag) will pivot the view. The 2D controls available when viewing this windows are as listed here (but many 2D keys do not currently work)

Action	2D	VR
Use/Grab	Mouse left button	Right controller trigger
Rotate camera	Mouse right button (hold)	Move your head
Translate axes	Keypad 1 *disabled*	Wrist menu → Edit
Scale axes	Keypad 2 *disabled*	Wrist menu → Edit
Rotate axes	Keypad 3 *disabled*	Wrist menu → Edit
Move forward	W or up-arrow	Left controller touchpad up
Move backward	S or down-arrow	Left controller touchpad down
Move left	A or left-arrow	Left controller touchpad left
Move right	D or right-arrow	Left controller touchpad right
Wiring mode	Keypad 5 or O *disabled*	Wrist menu → Wire/Code
Play mode	Keypad 0 or P *disabled*	Wrist menu → Play
Create object	via console *disabled*	Wrist menu → Store/Add
Save Zone	F8	Wrist menu → Home/Save

Starting VR

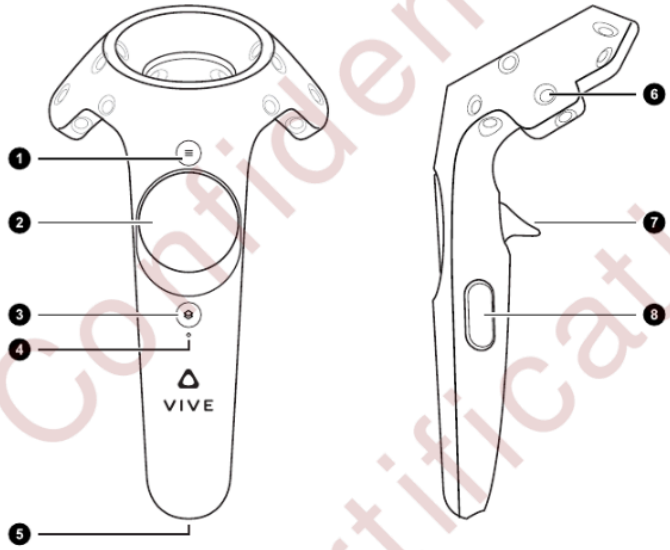
Press the F5 function key and you can then don the HMD (headset) and you will be in a 3D space. Pick up the controllers in both hands.

The left-hand controller is the one with a sideways wrist menu containing 5 items.

Use the left-hand controller trackpad to move forward, back or left/right. You fly where you are looking.

The right-hand controller ray selects stuff like the buttons on the wrist menu. Use the trigger to select an item. **PLAY** is the normal, default mode. The other modes enable you to make changes.

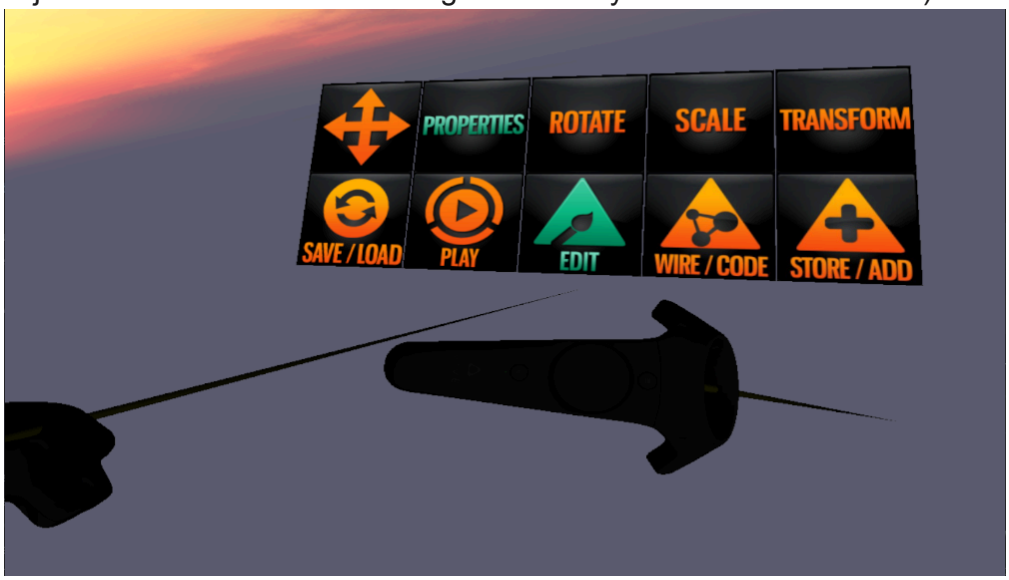
Controls on a controller



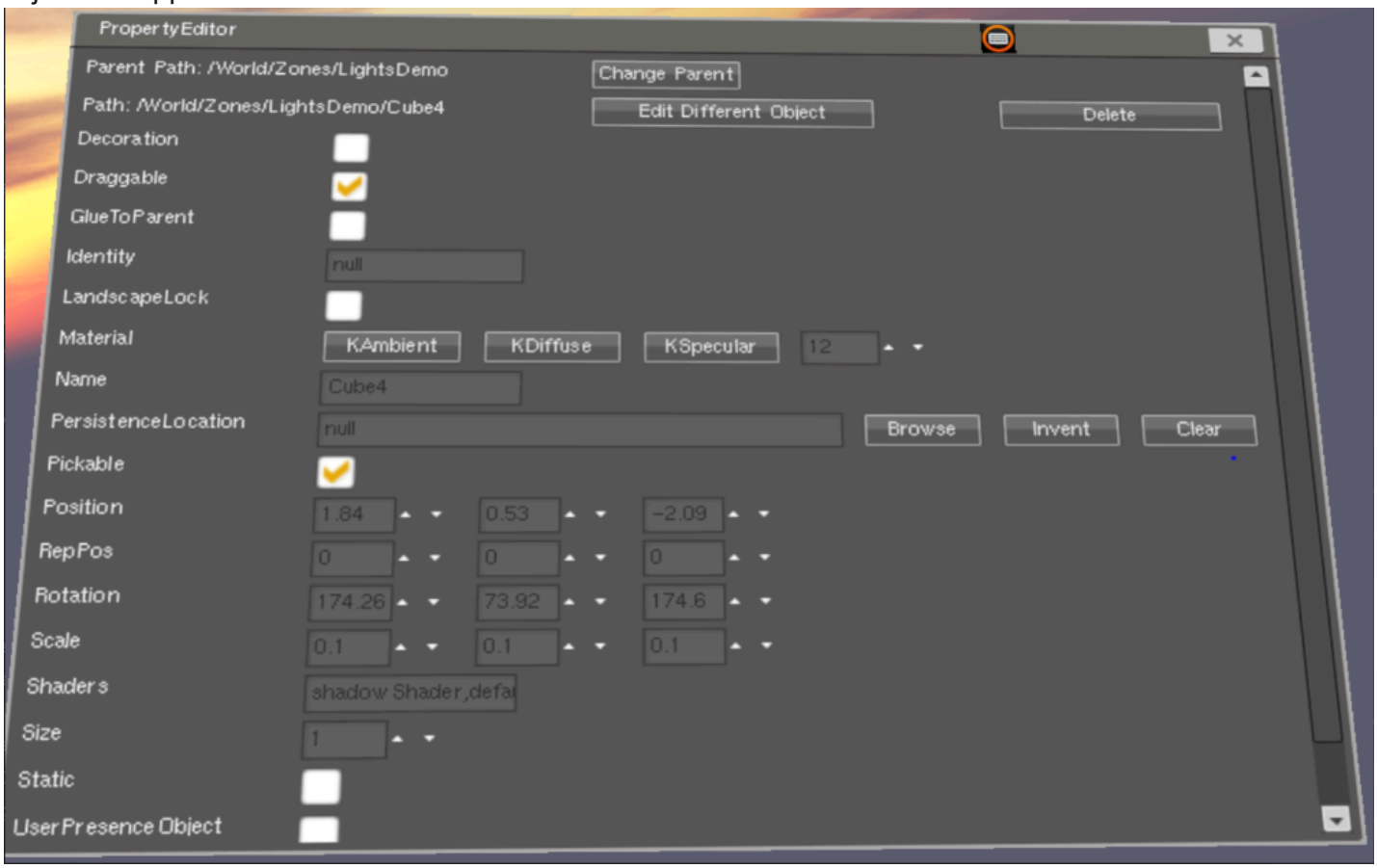
1	Menu button
2	Trackpad
3	System button
4	Status light
5	Micro-USB port
6	Tracking sensor
7	Trigger
8	Grip button

You can pick items up and move them by pointing the ray at the item with the right-hand controller and holding the controller trigger to select it.

(you have to be in PLAY mode or EDIT mode with the four arrows icon selected to move objects - the PLAY menu icon is green when you are in PLAY mode.)



Choose the **EDIT** menu item and then the **PROPERTIES** item to enable the Property mode. Then point at an object and use the trigger to select it. The grey Properties Editor for that object will appear on the left wrist.



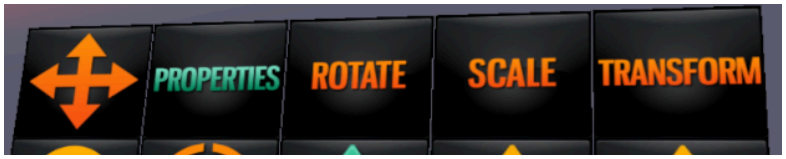
You can then change values on this form to alter an object. You can **Name** the object.

The **Material** (especially **kDiffuse**) changes the material color of the object.
KAmbient is the color reflected by ambient light. **KSpecular** is the color reflected by specular light. The number after the color buttons is **Shininess** -sharpness of the specular reflection.

Position, **Rotation** and **Scale** change those values numerically.
The **Draggable** checkbox enables a user to move the object in PLAY mode.
PersistenceLocation is where an object is stored. If it is null, then it is stored by a higher-level object which contains it.

The icon at the upper right is Display Keyboard (to enable typing in words and numbers)

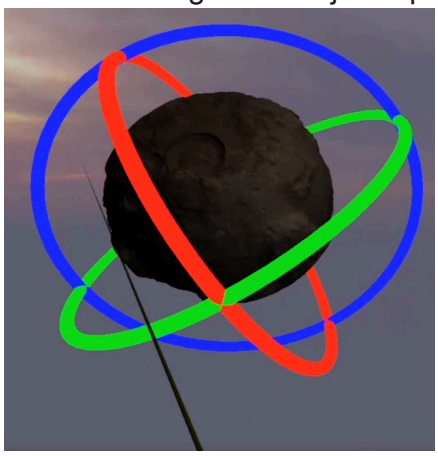
The **Edit Different Object** button displays an Object Browser window to see a list of all objects in the Zone. The **Delete** button will remove/destroy the object.
Point at the **X** in the upper right to close the Properties Editor window.



The 4-direction arrows menu item puts you in a mode that allows you to grab an object and freely move and rotate it. However, if an object's **Draggable** property is unchecked, it can't be moved in this mode

The **ROTATE**, **SCALE** and **TRANSFORM** menu items alter an object more specifically.
Use ray/trigger to select the menu item, then select the object you want to alter.
Bright colored arrows will appear around the object.
Point at an arrow, hold the trigger and move to change the object.

ROTATE changes the object's pitch,yaw and roll angles.



SCALE changes the size of the object in X,Y,Z.
The bar with a cube on the end changes the overall size of all three axes of the object, the other three arrows only change the size along one axis.

TRANSFORM moves the object's position by X,Y,Z.

Move the colored arrow to move the object

Adding Content to Your Zone

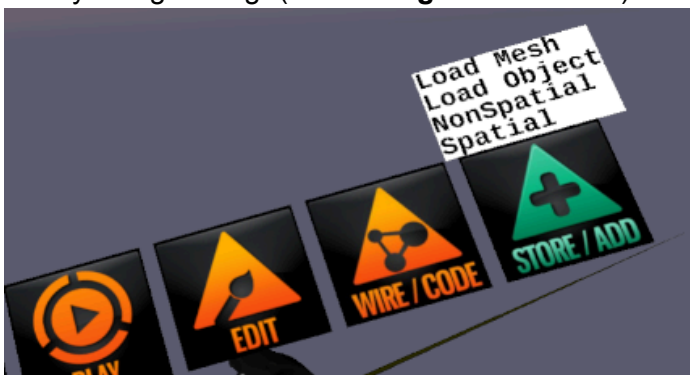
You can add new objects to your zone.
Things (objects) have a Name and a Class (type of object). Some example classes are: Light, Mesh, RanVec3, Cube, Torus, ToggleButton, Timer, Skybox, FloatValue, Cone, Arrow, Composite

A Composite can “contain” child objects (groups of objects), typically mesh objects

Most complex shapes (visible objects) are a Mesh (a 3D model)

Objects have a bunch of properties such as position, rotation, scale, related filenames
Material: color in RGB in ambient, diffuse and specular, shininess and reflectivity. You can view and change these in the Property Editor.

Some objects can send messages, like when you push a button, it sends a message. A message is sent to a Receiver object. These are detailed in an object’s Outputs property, and can be put to use by doing “wiring” (see **Wiring** section below)



Click the **STORE/ADD** icon on the menu.
A white cascading hierarchical menu will appear. As you point at an item, a submenu of additional related choices will appear to the right. There are four kinds of objects.

- **Load Mesh** - loads a 3D model file from your local disk.
- **Load Object** - loads a .CLO file that is not a Zone from local disk
- **NonSpatial** - perform calculations and control flow
- **Spatial** - Objects that are positioned in your Zone

For example, select **STORE/ADD** mode then select **Spatial**, then select **Visual**, then select **Cube** and a new cube will appear in your zone..

Load Mesh - displays a file browser that you use to locate a disk file of a model you want to import. (currently only supports Collada .dae format 3D model files) Select the file and the object appears. See the section “Adding external assets” further down in this document
Load Object - displays an object browser you may use to load a .clo file that contains objects.

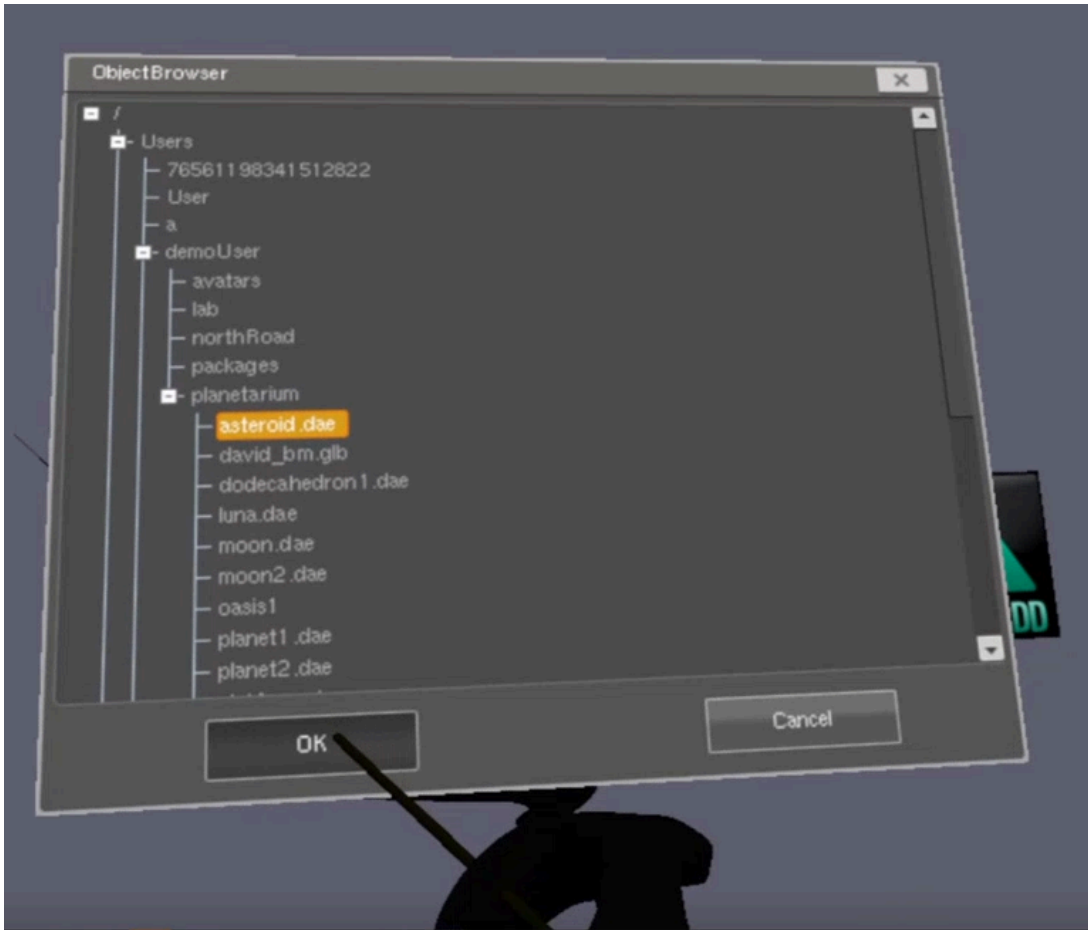
Adding external assets

3D objects can be “assets” created with regular 3D modeling tools such as Blender, Cinema4D, Maya/3DS or Sketchup. These tools create the meshes (the shapes) and color or texture the materials. Pre-made 3D model assets can be downloaded online from Turbo Squid and many other websites.
Check asset files before loading a mesh (different 3D model file format files can be checked, but must be converted to .dae file format to load into Monkey):

- Open it in Blender.. Verify that it looks ok, no error reported. Export to DAE if needed.
- Open file in Open3DMod. You can verify that the model can be loaded and that the textures/materials are correct. You can then export it as Collada .dae if it is not already a dae file.. Notice any error messages.
- Now you are ready to load it into Monkey.

The file must be placed in a folder within your .cldata/Users folder for Monkey to be able to find it. The model may then be loaded into Monkey to create a VR object.

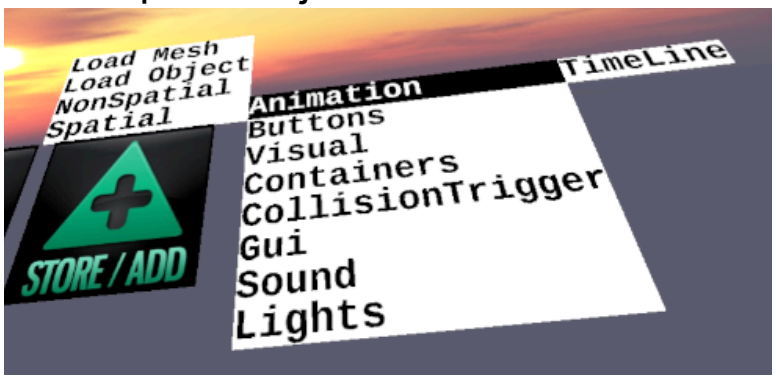
Use the **Load Mesh** menu item to bring a 3D model .dae file object into your Zone from your local disk. (you must place it must be within your .cldata/Users folder) You will see a file browser window to choose the .dae 3D object file from.



Spatial objects

Spatial->Visual - are objects you can see.

Other spatial objects



Animation->timeline - needs to be wired to things to be useful
Buttons
MomentaryButton
ToggleButton

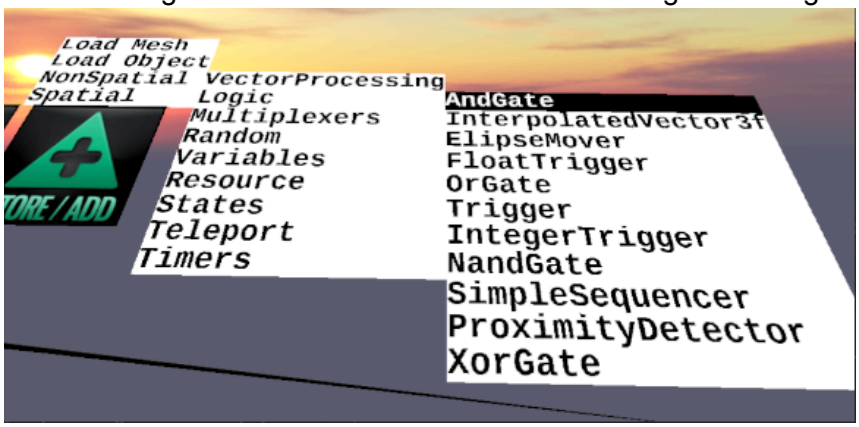
Containers->Composite - for grouping objects
CollisionTrigger - sends a message when an object collides with it - needs to be wired to things to be useful
Gui->GuiKeyboard - displays a keyboard to type on - needs to be wired to things to be useful

Sound - add an object that plays audio
Lights - objects that emit light to illuminate the scene

NonSpatial objects

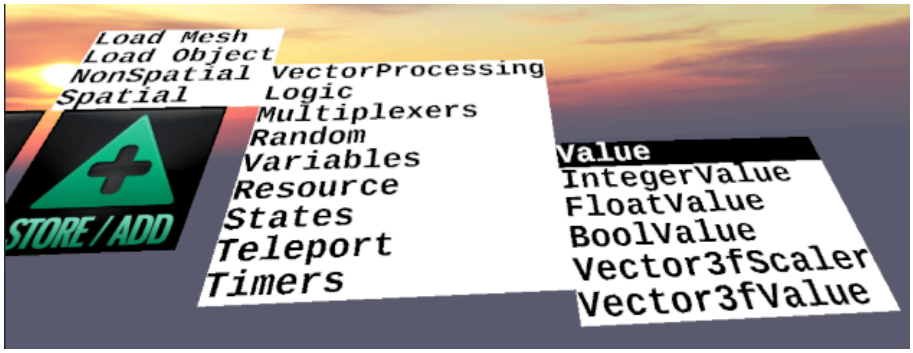
These objects are for “programming” making objects interactive.

VectorProcessing->offset
A vector is a set of 3 related numbers, typically an X,Y,Z position, or rotation specification. Offset allows adding one vector to another to do something like change the position.



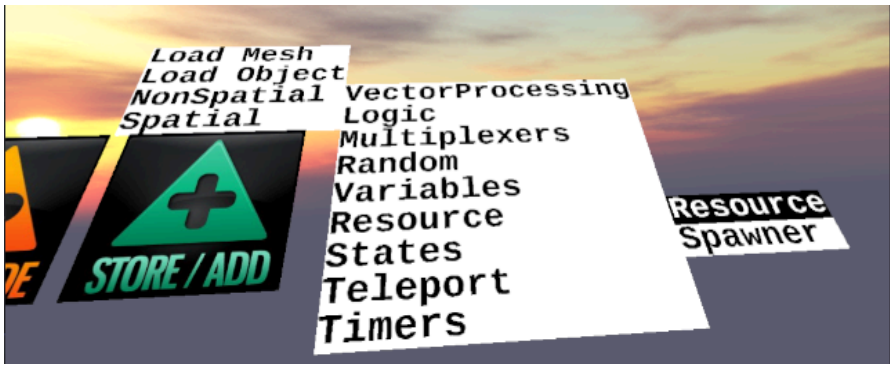
Logic objects
AndGate
InterpolationVector3f - given two vectors,and an amount float that ranges from zero to one, this outputs an altered vector that is interpolated between the two vectors based on the amount.
ElipseMover - changes positions around an ellipse
FloatTrigger - fires a message when a float number is equal to another
OrGate
Trigger - fires a message
IntegerTrigger- fires a message when an integer number is equal to another
OrGate
NandGate
SimpleSequencer - an object that steps through a number of outputs
ProximityDetector - sends a message when an object gets close to another object
XorGate

Multiplexers->Vector3fMutiplexer - unpacks a vector into 3 individual values
Random-->RanFloat - random number
Random->RanVec3

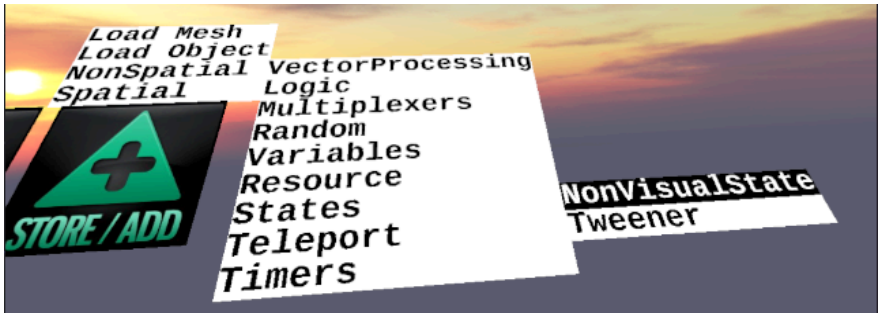


Variables objects
Value

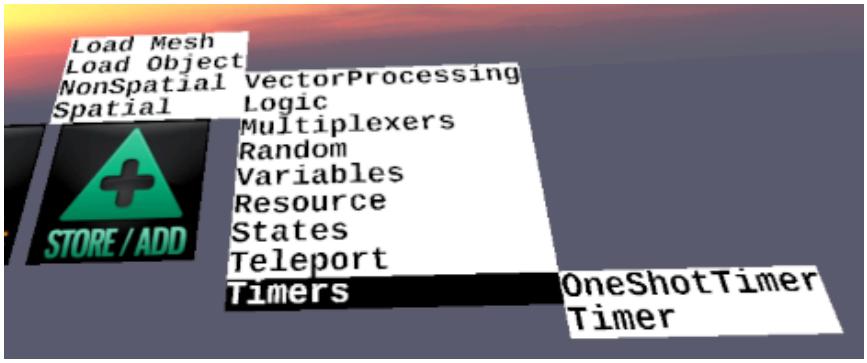
IntegerValue
FloatValue
BoolValue
Vector3fScaler
Vector3fValue



Resource objects
Resource
Spawner

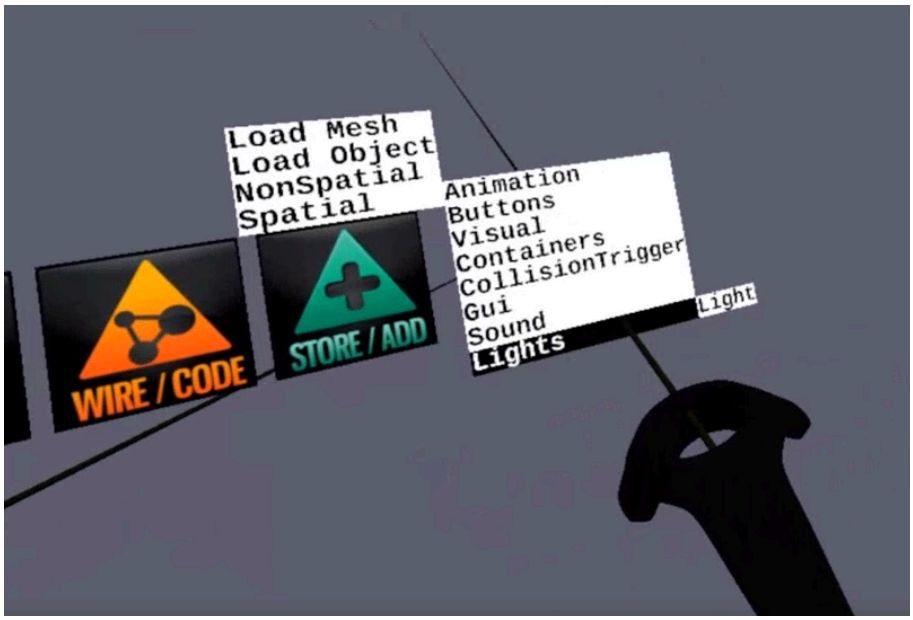


States objects
NonVisualState
Tweener
Teleport->Teleport
Contains a destination location, orientation and zone. When triggered, it moves the user to that destination

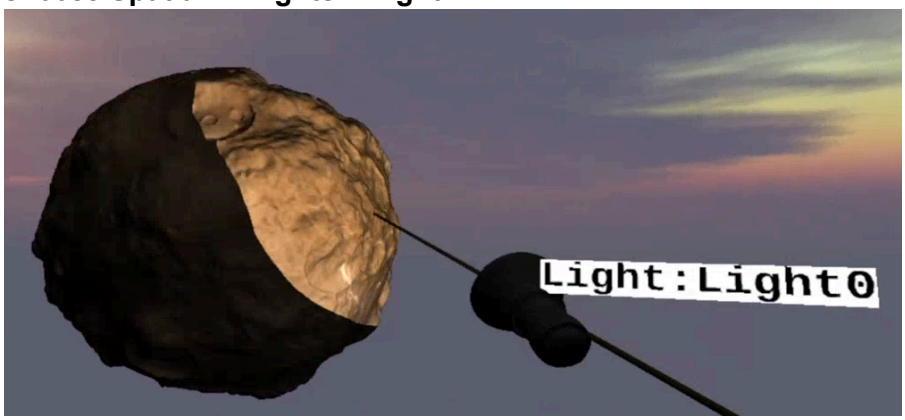


Timer objects
OneShotTimer - sends a message after a delay
Timer - sends periodic messages

Adding a light



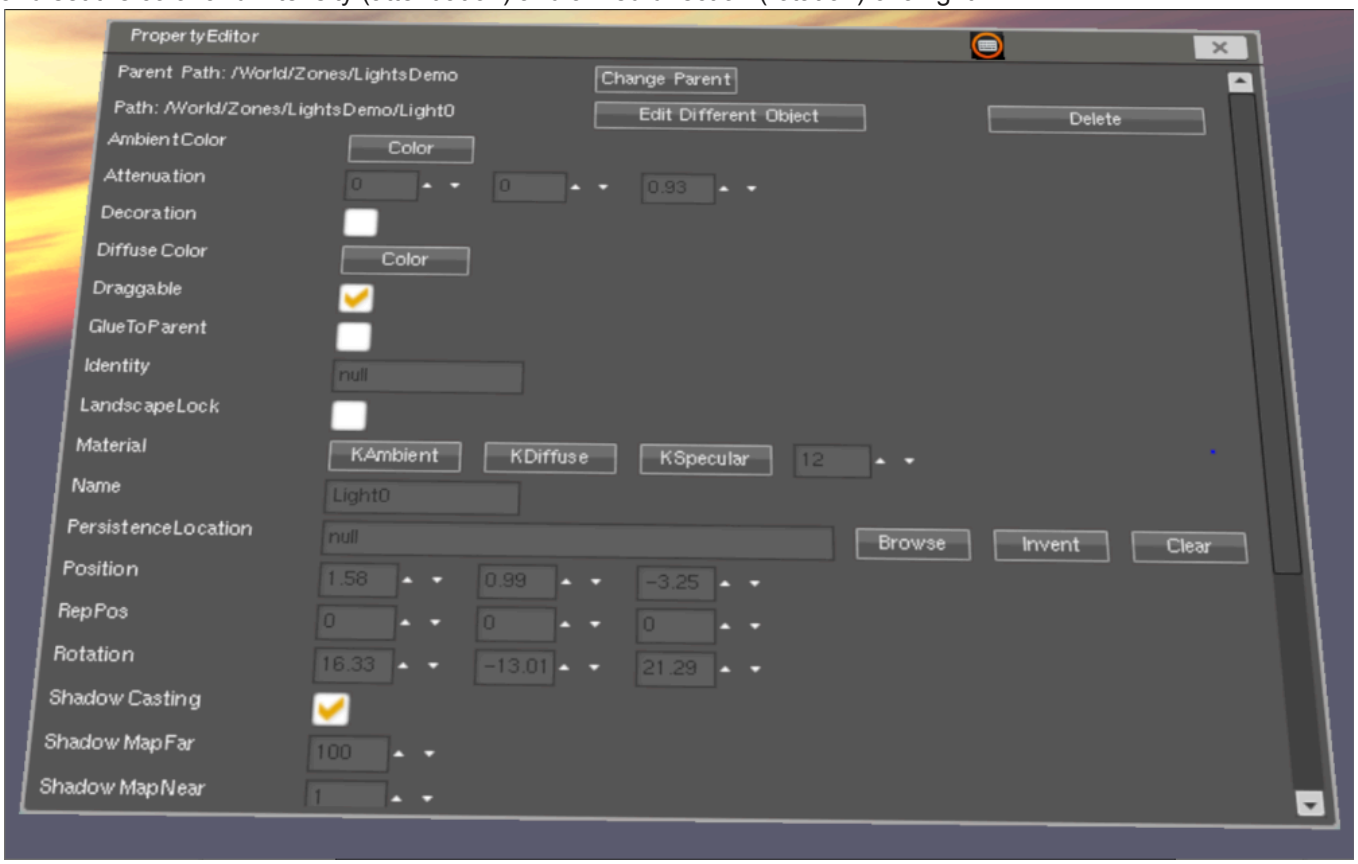
Choose **Spatial** -> **Lights** -> **light**



A spotlight is added.

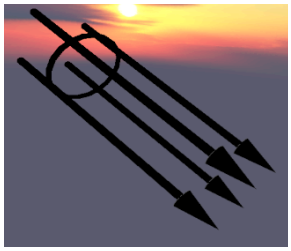
In the **Property Editor** for the light, you can choose 4 different types of lights (the default is a spotlight)

and set the color and intensity (attenuation) and aimed direction (rotation) of a light



SPOT - shines a cone of light in a specific direction. You can specify the radius of the cone. Its intensity fades out (falloff) over distance. Adjust the amount of falloff with the third Attenuation value (set this by entering a value between 0 and 1)
Set the AmbientColor to 0,0,0 so that only the cone of light is illuminated.
DiffuseColor sets the color of the cone of light.

POINT - shines in all directions, from its location, like a light bulb. Same falloff as a spotlight.

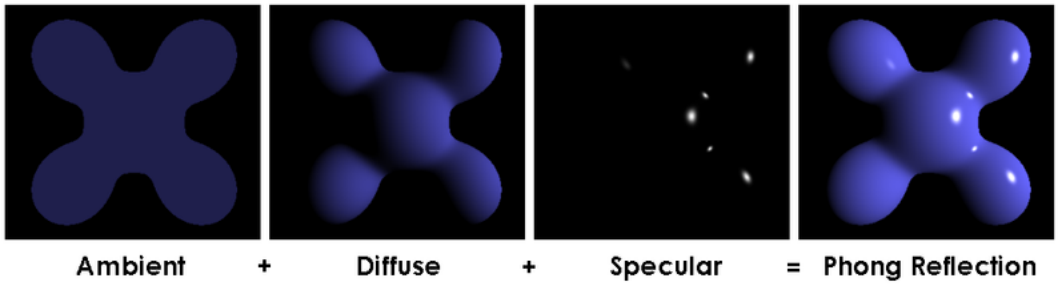


DIRECTIONAL - shines parallel rays like the sun. It does not fade out over distance

AMBIENT - illuminates from all directions, sort of like from the sky acts like a huge light. It does not fade out over distance.

Color of light

- **Ambient lighting**: even when it is dark there is usually still some light somewhere in the world (the moon, a distant light) so objects are almost never completely dark. To simulate this we use an ambient lighting constant that always has the object reflecting some color.
- **Diffuse lighting**: simulates the directional impact a light object has on an object. This is the most visually significant component of the lighting model. The more a part of an object faces the light source, the brighter it becomes.
- **Specular lighting**: simulates the bright spot of a light that appears on shiny objects. Specular highlights are more inclined to the color of the light than the color of the object.

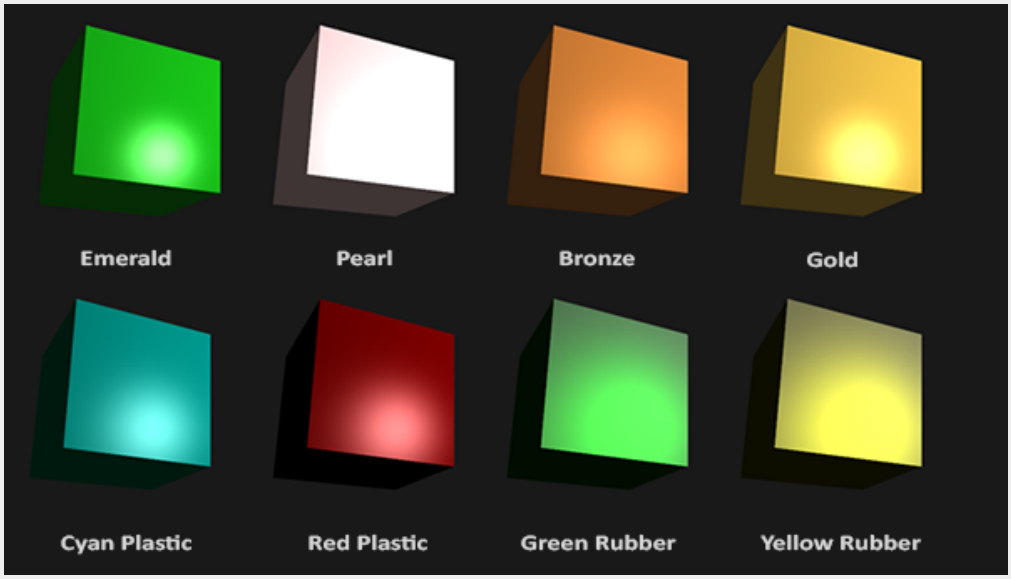


The ambient light is usually set to a low intensity because we don't want the ambient color to be too dominant. The diffuse component of a light source is usually set to the exact color we'd like a light to have; often a bright white color. The specular component is usually kept shining at full intensity. All three combine to become Phong reflection

Color of materials

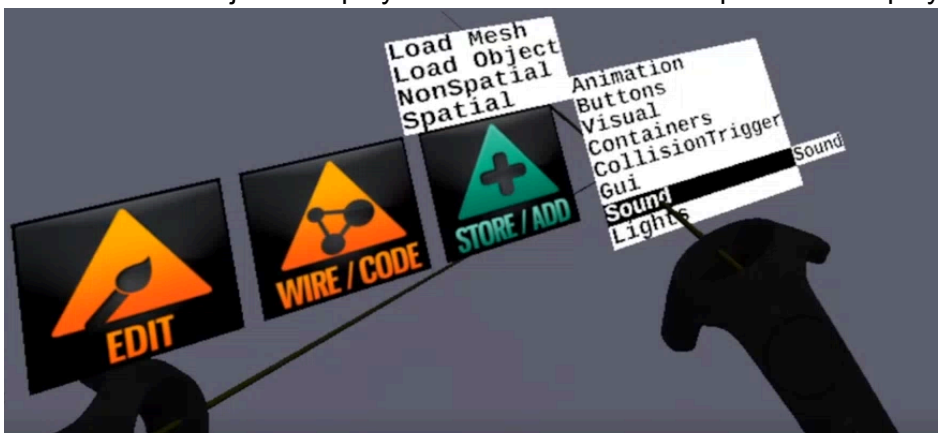
In the real world, each object has a different reaction to light. Steel objects are often shinier than a clay vase for example and a wooden container doesn't react the same to light as a steel container. Some objects reflect the light without much scattering resulting in small specular highlights and others scatter a lot giving the highlight a larger radius.

The `ambient` material defines what color the surface reflects under ambient lighting; this is usually the same as the surface's color. The `diffuse` material defines the color of the surface under diffuse lighting. The diffuse color is (just like ambient lighting) set to the desired surface's color. The `specular` sets the color of the specular highlight on the surface (or possibly even reflect a surface-specific color). Lastly, the `shininess` impacts the scattering/radius of the specular highlight.
Example materials settings: http://devernay.free.fr/cours/opengl/materials.html



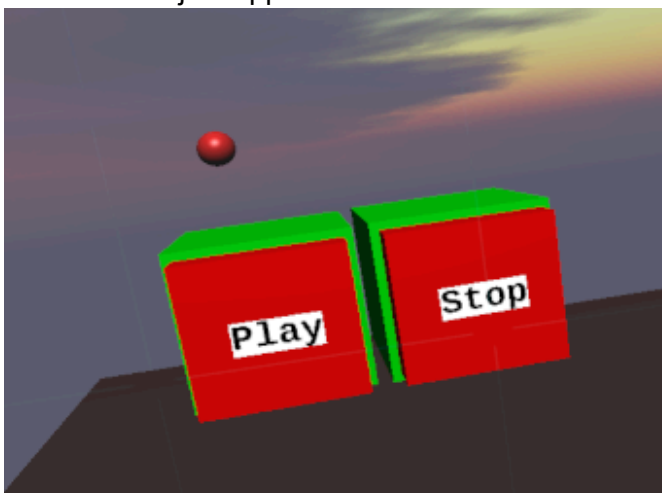
Adding a sound object and wiring it up

We will add an object that plays an audio file and wire up controls to play and stop the audio.

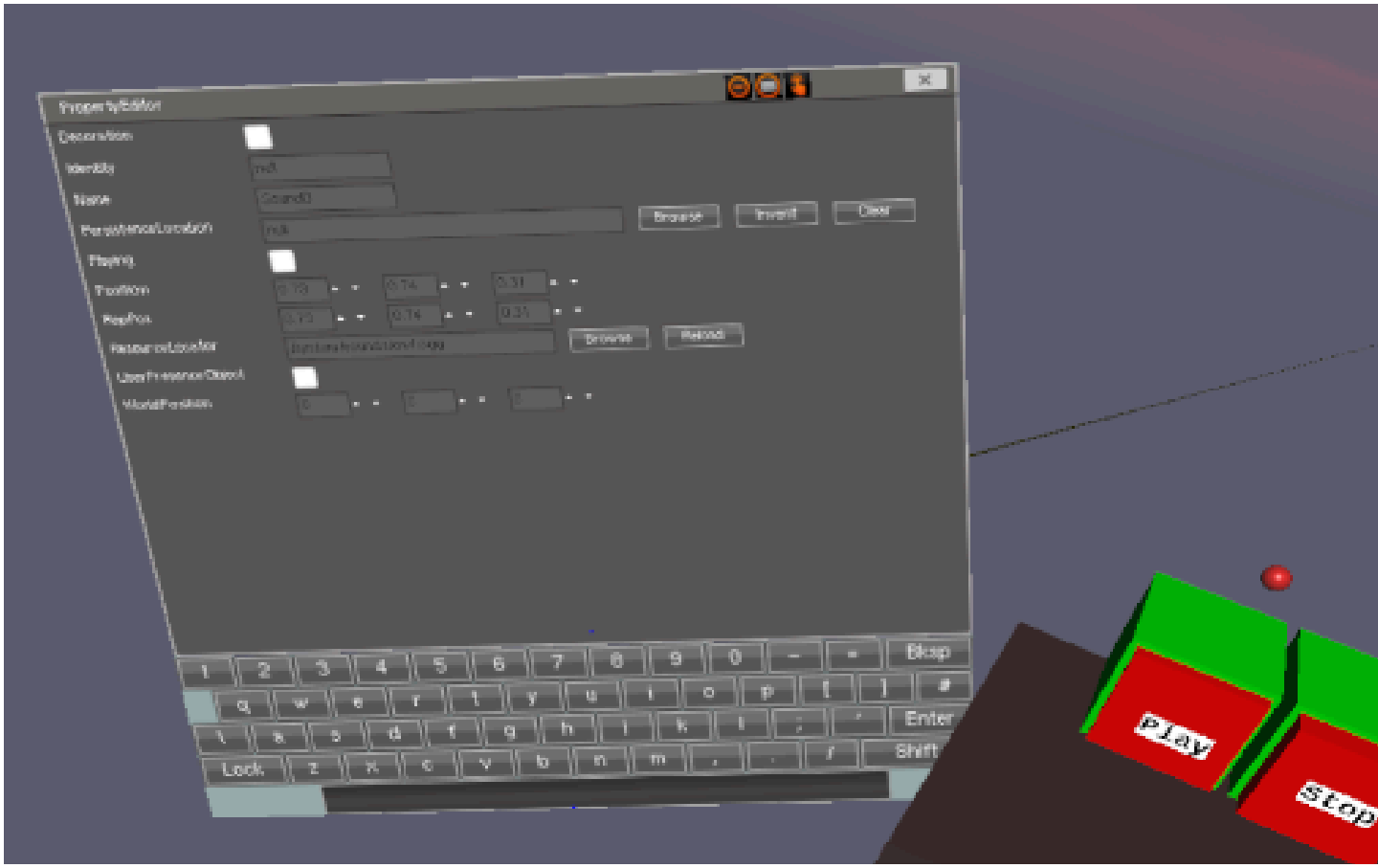


Choose **STORE/ADD** menu item and then **Spatial** -> **Sound** -> **Sound**

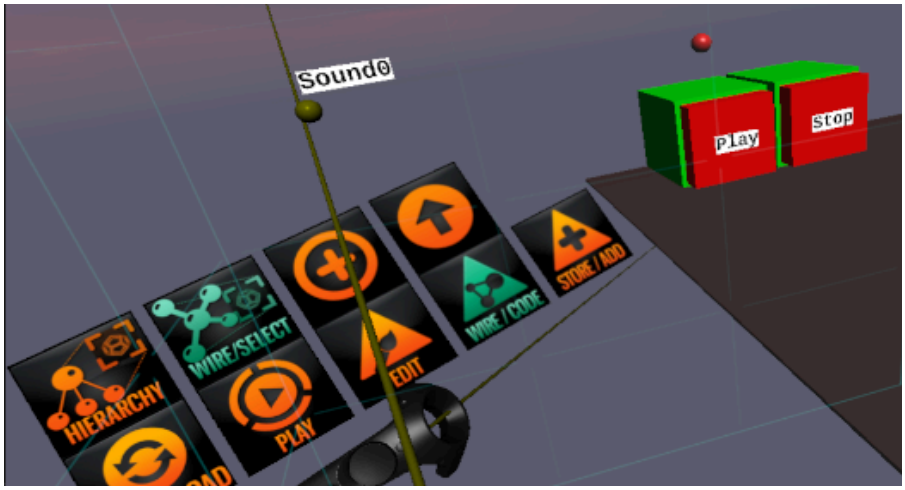
The sound object appears. It looks like a red ball.



Select it to assign a sound file to it in the Property Editor. (there is already a default sound file provided)

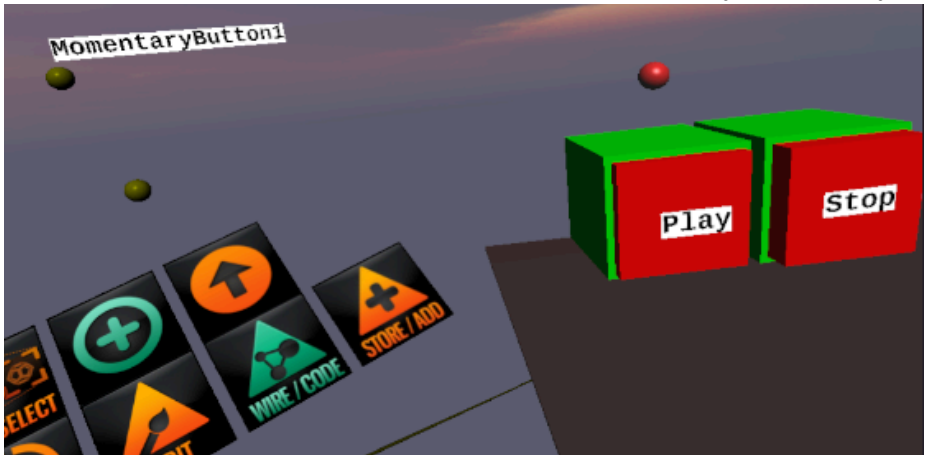


Two momentaryButtons have previously been placed in the scene and labeled Play and Stop.
We are going to wire them up to control the sound.

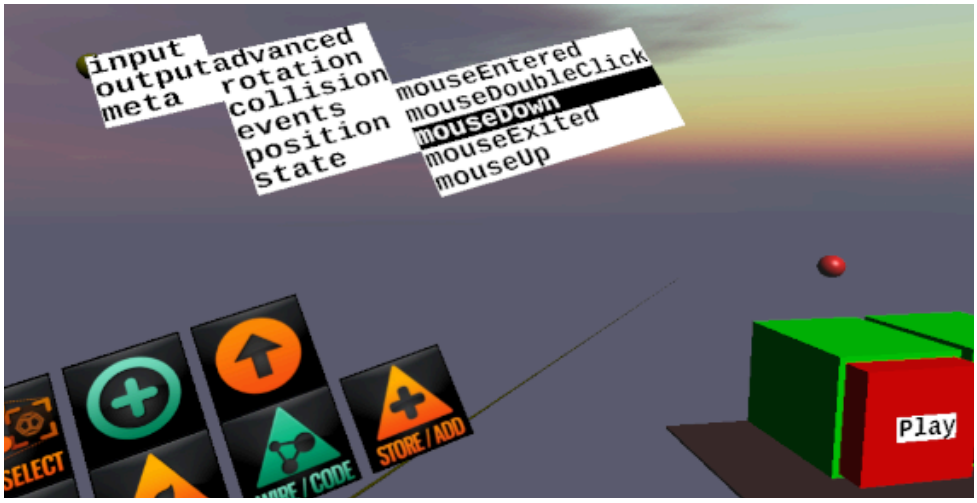


Choose the **WIRE/CODE** menu item. Choose the **WIRE/SELECT** item. Click on the red Sound Object and a new dark green ball will appear floating over the menu. If you point at it, a label “Sound0” will appear. This is the wiring ball for the sound object.

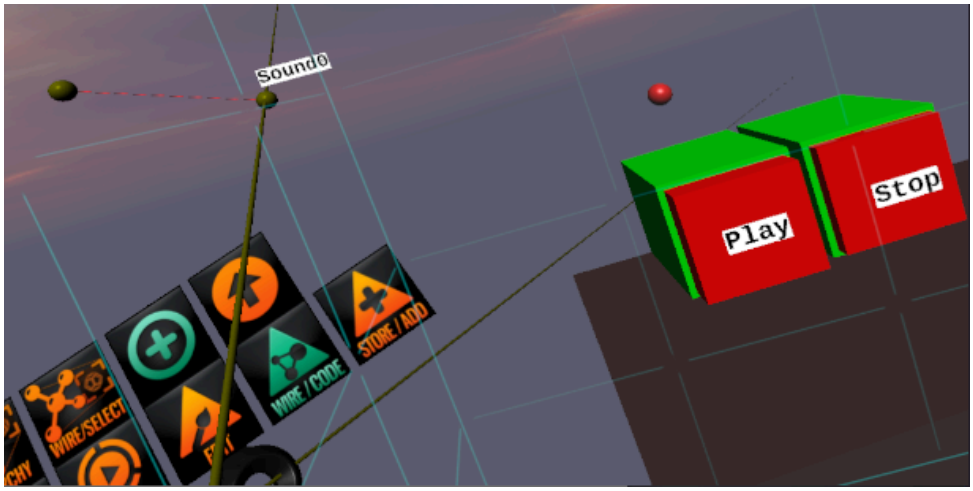
Now we are going to connect the play momentaryButton to control the sound.
Click the **+** (in a circle) menu item .Then click on the Play momentaryButton.



A second ball will appear hidden behind the first dark green ball. Grab a ball and move it a little ways away so you can see two dark green balls. Each will have a label you can see if you point at it.

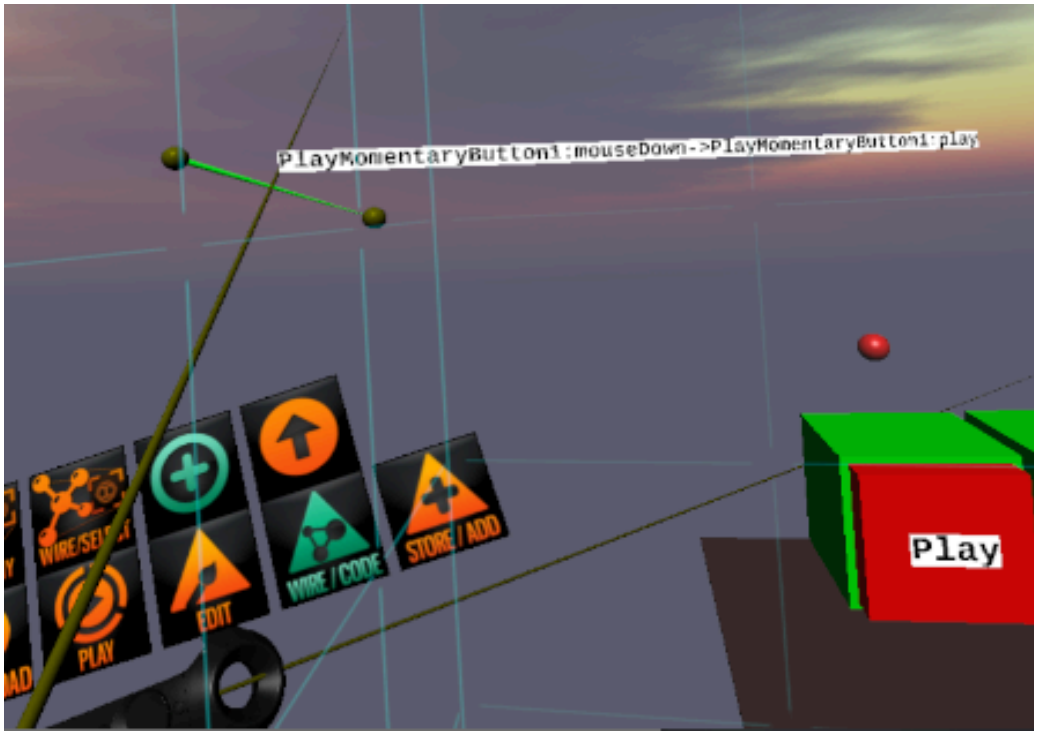


Click on the + menu item again to allow you to choose messages. Point at the ball for the Play momentaryButton and a menu of messages will appear. Choose the **output->events-> mouseDown** message.

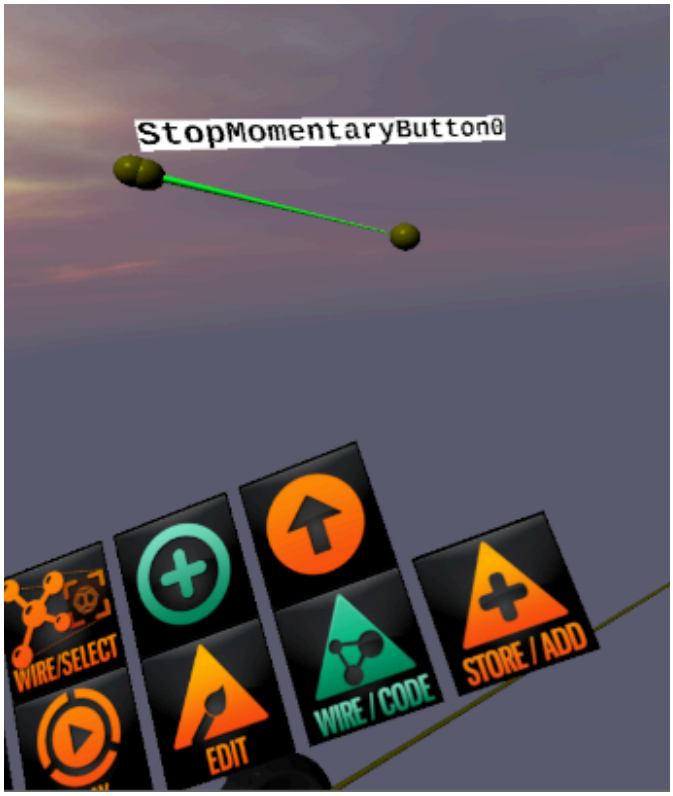


A red line will appear. You connect that red line to the dark green sound ball, by clicking on that ball.

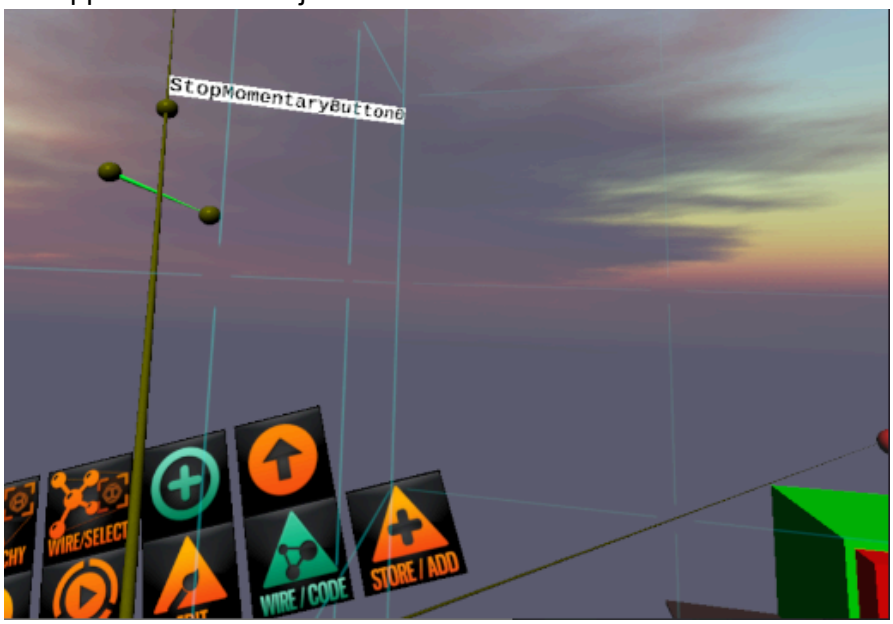
A menu of messages for the sound ball will appear. Choose the **input->control->play** item



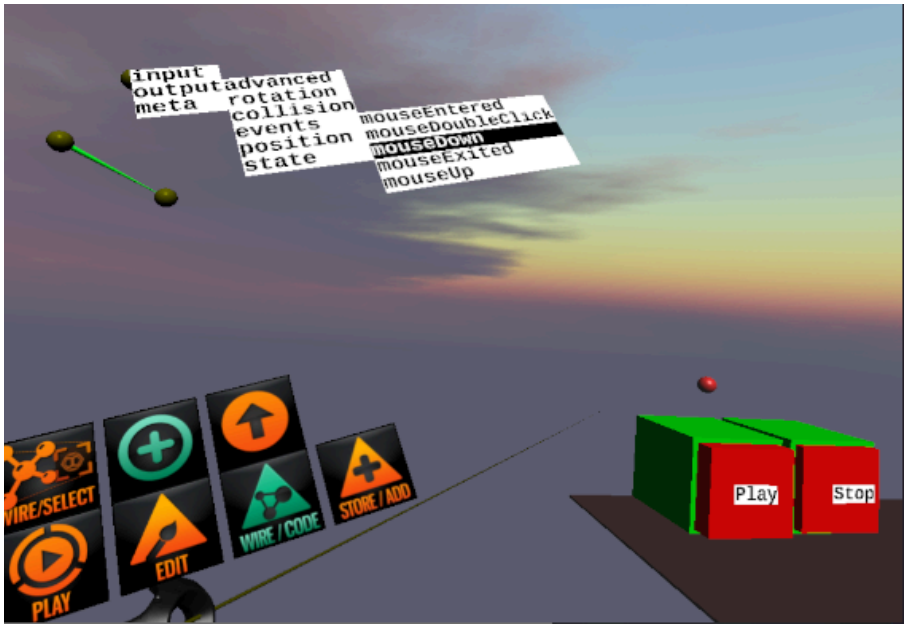
Now a green line shows that the two objects are wired together and when a mouseDown event happens on the playMomentaryButton it sends a message to the sound object to play.



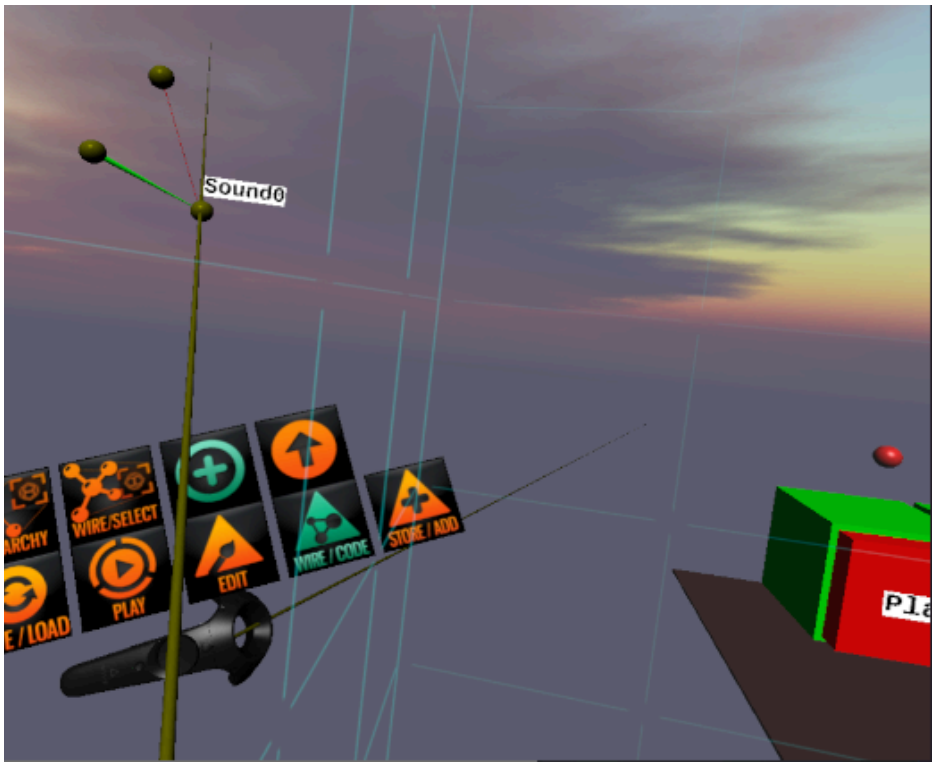
Now click on the + menu item and then click on the Stop momentaryButton. Another dark green ball appears for that object.



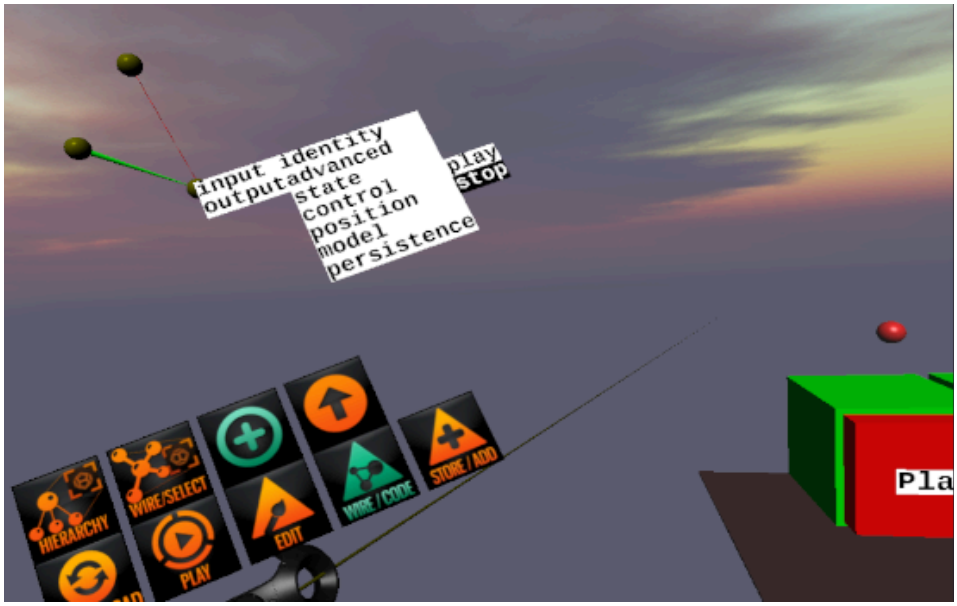
Move the new ball a bit away from the others.



Click on the + menu item again and then click on the ball for the Stop momentaryButton to get the messages menu. Choose output->events->mouseDown



A red line appears. Click on the sound ball.



The message menu appears, Choose **input->control->stop**.

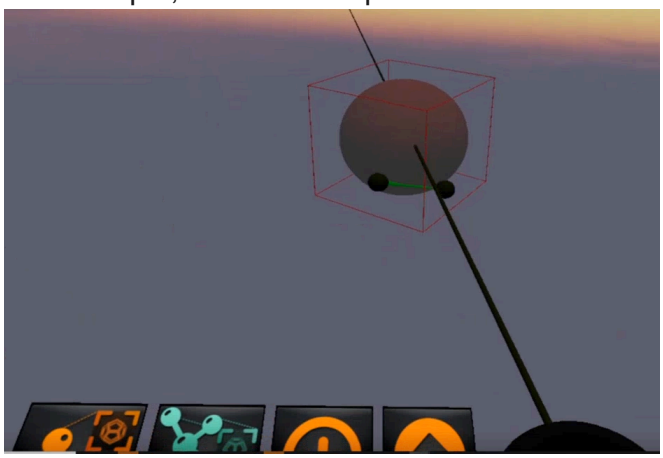
Now the Stop button is wired to tell the sound to stop playing.

The wiring is complete. Click on the **PLAY** menu item to “go live” and then you can use the Pay and Stop momentaryButtons in the scene to start and stop the audio playback.

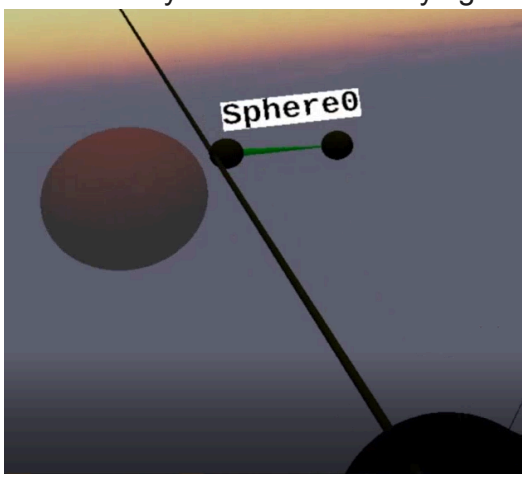
Wiring

You can go into WIRE mode by choosing the **WIRE/CODE** menu item

When you are in wire mode, you get the wiring diagram for anything you click on, just like with properties.The wiring looks like dark green balls connected by green lines, sort of like a chemistry molecular model.
For example, click on the sphere



you get a little wiring diagram on the wrist. you can drag it around like anything else
hover the ray over it to label saying what it is

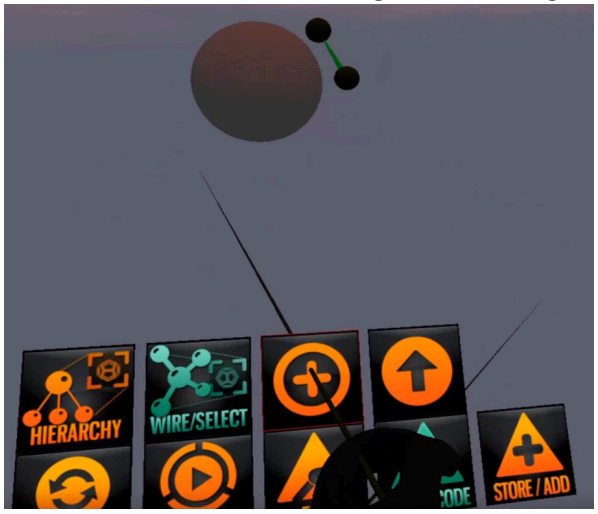


if you ray, click and release with the right controller trigger, that will bring up a menu, you can use to wire objects together

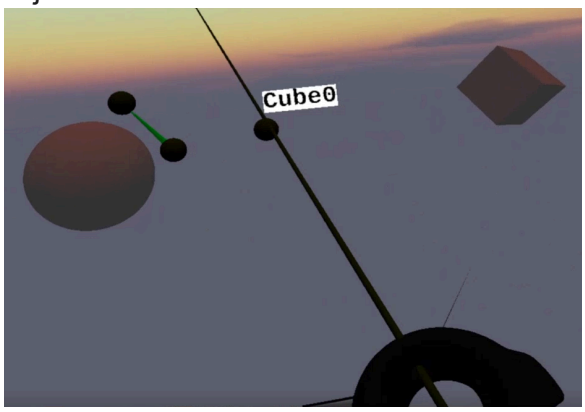
Parameters:

if a wire is red it needs parameters.
but you can't put them inside the wire, you have to get them from another object for now

the + button adds something to the wiring, for example another object: the cube.

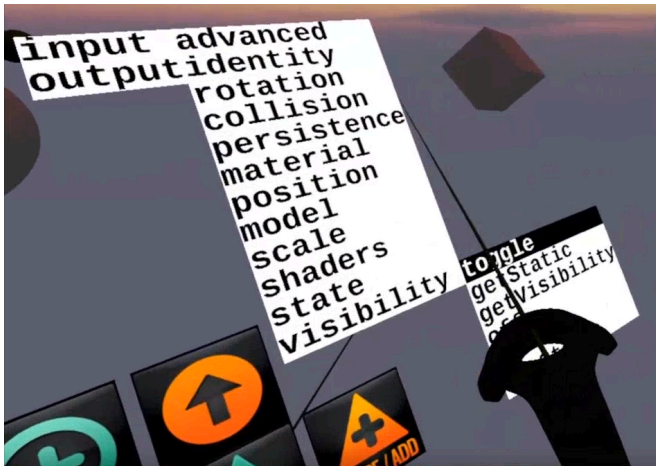


for example click on another another object such as the cube and a new ball appears for that object

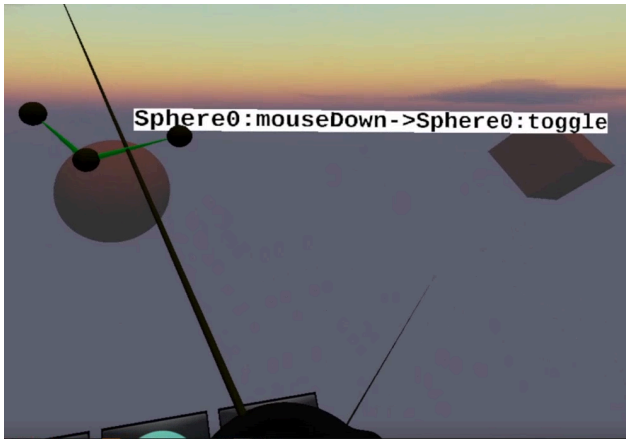


Click on the ball and you can access the messages that object can **Output** (send). In this case: one of the **events**: **mouseDown**.

Now we are going to connect that wire to the cube's **input, visibility, toggle.** This will make the visibility property of the cube toggle on and off when the sphere is clicked on.



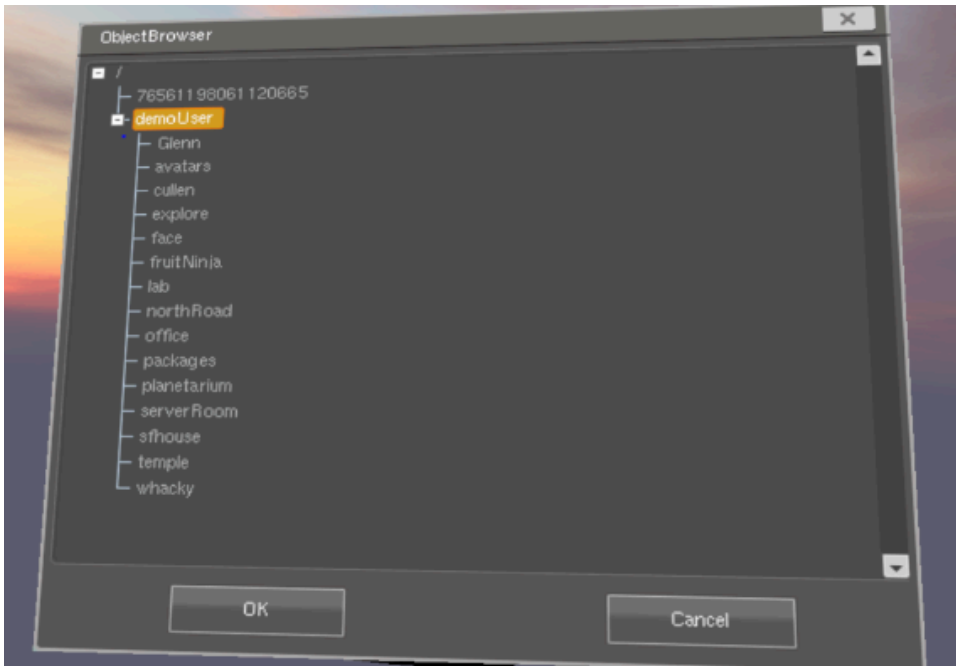
Point at the wire to see what it does:



Saving and Loading Zones

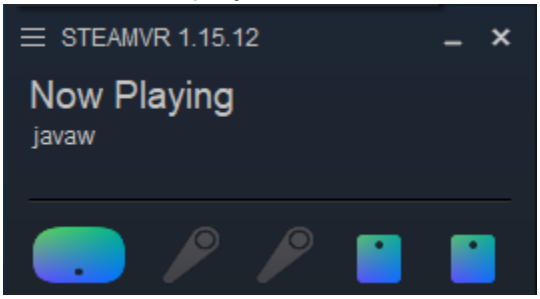
Use the SAVE/LOAD -> **SAVE** menu to save the current Zone on disk.

HOME brings you to the STEAMVR Home environment.
OPEN brings up the ObjectBrowser so you can select another zone on your disk to switch to.



Screen Capture

Use the hamburger menu (three horizontal lines in the upper left) on the STEAMVR window to turn on "Display VR view"



That will give you a window on the screen which you can make larger.
Then use the keyboard **Ctrl-ALT-PrintScrn** to capture the image of the window onto the clipboard.
You can then paste it anywhere (like into the Paint app)

*If this is not working, find the file **steamvr.vrsettings** in C:\Program Files (x86)\Steam\config and delete it.*

In the small STEAMVR window, hamburger menu, Choose “Developer” and then “Debug Commands”. It shows a list of keyboard commands. One of the commands is screenshot_request (press the s key) that will capture a stereo screenshot .PNG file into:
`C:\Program Files (x86)\Steam\steamapps\common\SteamVR\screenshots`

Storage of content

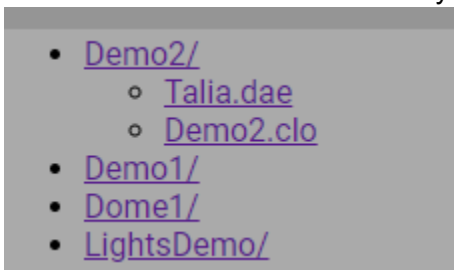
The .clData/Users/<username> folder in your Windows home directory contains the data for each zone in a subfolder. You will find a whole filesystem in there replicated from the AWS S3 cloud.

.clo files ("Constructive Labs Object") - contain JSON text data. If you put that into an online JSON pretty-printer you can read the content for the zone.
Properties which do not deviate from the default are not included - just overridden things are included in the clo file.

The **.dae** files contain 3D model geometry These are a COLLADA XML file. They reference textures, in other .jpg and .png files. Here's a useful ool to inspect .dae files:
https://github.com/dabinn/open3mod-kai/releases/download/1.2/open3mod_1_2_x64_standalone.rar

Sharing your Zone

Zones you create are automatically stored (synced) in the AWS S3 cloud.
Login to https://portal.constructivelabs.com/
Go into Content. Select the Zone you want to share:



Check the Read checkboxes for Group and World

click Apply