

Orbiter Mesh Tools for Blender 2.81 – Build and texture a simple ship.

January 18, 2020

For this tutorial we will use Blender and the Orbiter Mesh Tools plug-in to create a simple 'cube' vessel that uses an Orbiter vessel configuration file. This vessel will not do anything, but it will introduce you to some concepts needed to build and texture a vessel mesh file using Blender and the Orbiter Mesh Tools plug-in.

Assumptions:

- You have a basic understanding of Blender.
- Blender **2.81** is installed.
- Orbiter Blender Plugin **2.0.3** is installed. See the Readme.md in the git repository.
- Orbiter is installed in C:\Orbiter.

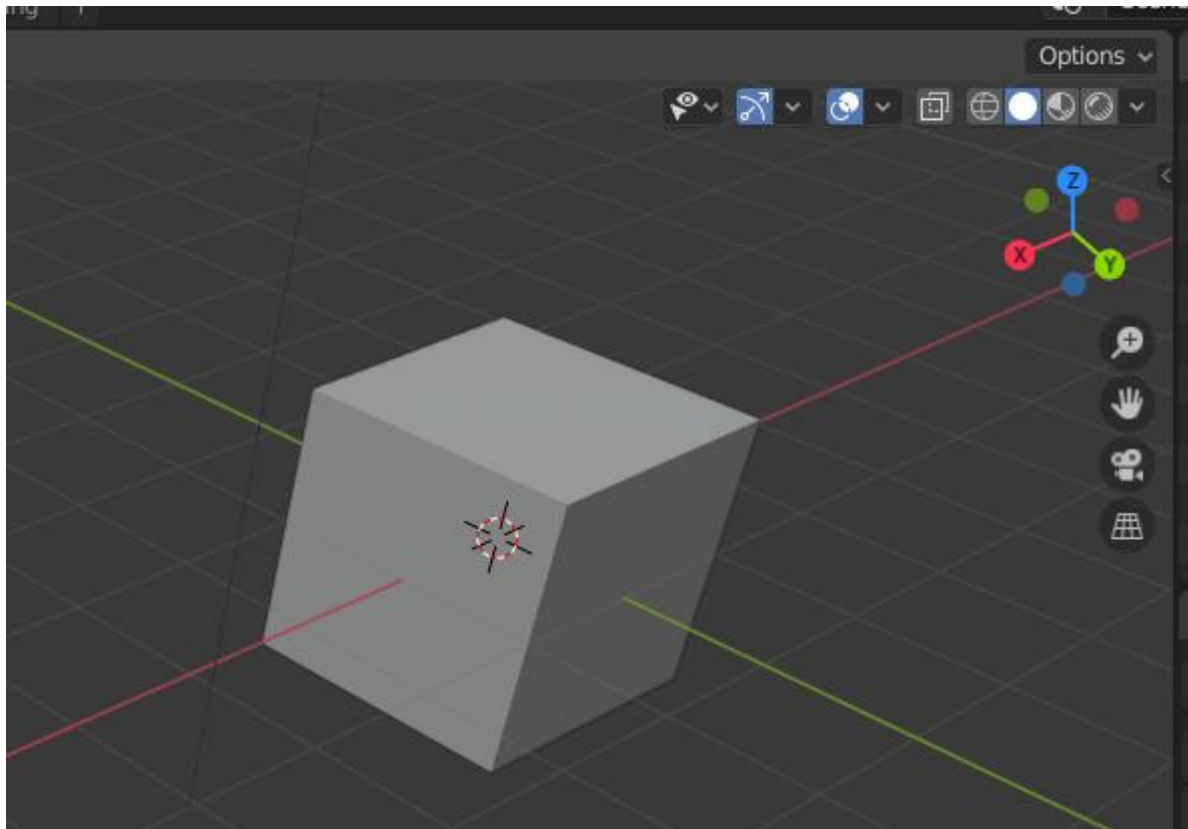
Create a Basic Mesh

Create a folder C:\MyShip. This will hold our blend file and any other files we may need.

When you first start Blender the default scene is comprised of three objects, a cube mesh, a camera, and a light. To keep things simple, we will use the default cube as our initial object. You can remove the camera and light if you wish, but they will not interfere with our model so we will just leave them for now.

A quick discussion of coordinate systems:

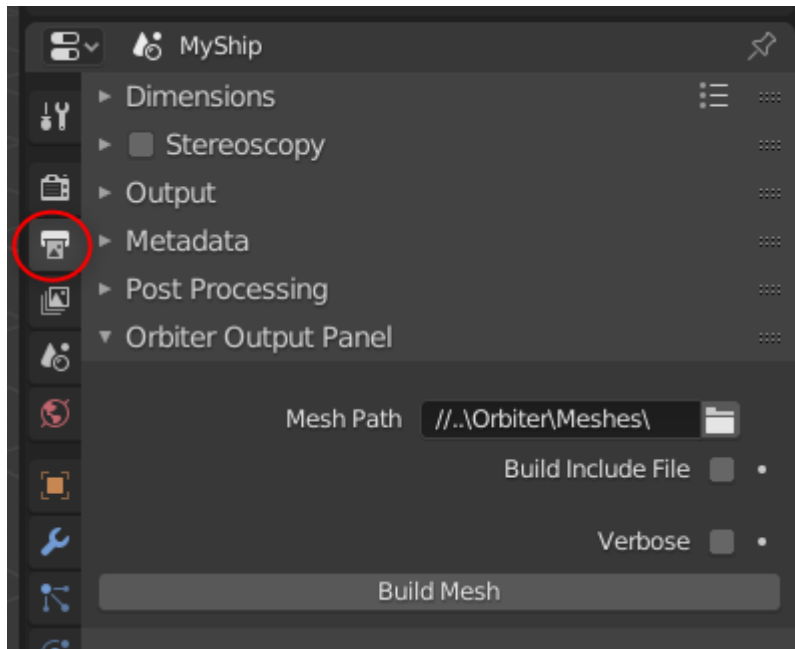
Orbiter uses a *left handed* coordinate system where positive *Z* is *forward*, positive *Y* is *up*, and positive *X* is *right*. Blender, on the other hand, is *right handed*. To make our mesh *left handed* Orbiter Mesh Tools swaps the *Y* and *Z* axis values when creating the mesh file. So, in Blender treat the *Y* axis as *forward*, and the *Z* axis as *up*. Doing this makes positive *X* values the *right* side of the vessel. Use the axis indicator in the upper right corner of the 3D View window to keep your directions straight.



In the above picture, the 'nose' of our vessel is pointing down to the right along the green line, so we are viewing the front, right, and top of the vessel.

Before doing anything save the default Blender scene to your working folder. Select menu: *File » Save As*. Navigate to the *C:\MyShip* folder. Name the file *MyShip*. Blender will add the blend extension. Your *MyShip.blend* file should now be in *C:\MyShip*. Before we can build the mesh file, we need to set some properties.

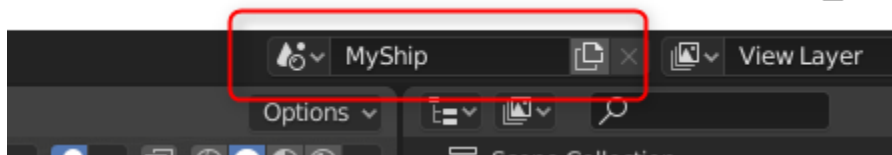
Select the *Output Properties* panel (the *printer* icon circled in red) and open the *Orbiter Render Panel*.



Set the Mesh Path: to `C:\\Orbiter\\Meshes\\`. (Blender will show the relative path from the blend file location)

Un-check *Build Include File*.

In the Blender *Scene* drop-down at the top of the window, rename the scene from *Scene* to *MyShip*.



You're now ready to build your first mesh file. Press *Build Mesh* to build the mesh file. The *Build Mesh* button is at the bottom of the panel where we set the Mesh Path. If everything worked, you will see a *Build Mesh Done* message appear briefly at the bottom of the Blender window.

At this point you should have a very simple *MyShip.msh* file in your *Orbiter Meshes* directory. Orbiter Mesh Tools will build a separate mesh file for every scene in the blend file. We have a single scene named *MyShip*, so we have just one mesh file.

Create an Orbiter vessel

Our ship will be based on a configuration file rather than a DLL so its functionality will be limited. We will setup the vessel configuration file next.

In `C:\\Orbiter\\Config\\Vessels` create a new text file named *MyShip.cfg*.

Using Notepad (or any other text editor) edit *MyShip.cfg* and put the following lines in:

```
; === Configuration file for vessel MyShip ===  
MeshName = MyShip  
Size = 1
```

Save and close the config file.

Next, we will need a scenario file that will use our ship. Our scenario will be very simple and will put our ship on a landing pad at Brighton Beach.

In *C:\Orbiter\Scenarios* create a new folder called *MyShip*.

Inside the new *MyShip* folder create a text file named *First.scn*. Edit *First.scn* in Notepad and enter the following:

```
BEGIN_DESC  
MyShip parked at Brighton Beach  
END_DESC
```

```
BEGIN_ENVIRONMENT  
System Sol  
Date MJD 52006.7491805055  
END_ENVIRONMENT
```

```
BEGIN_FOCUS  
Ship MyShip-01  
END_FOCUS
```

```
BEGIN_CAMERA  
TARGET MyShip-01  
MODE Extern  
POS 20.10 -145.00 -26.500  
TRACKMODE TargetRelative  
FOV 50.00  
END_CAMERA
```

```
BEGIN_HUD  
TYPE Surface  
END_HUD
```

```
BEGIN_MFD Left  
TYPE Launch  
NAV 0  
END_MFD
```

```
BEGIN_MFD Right  
TYPE Map  
REF Moon
```

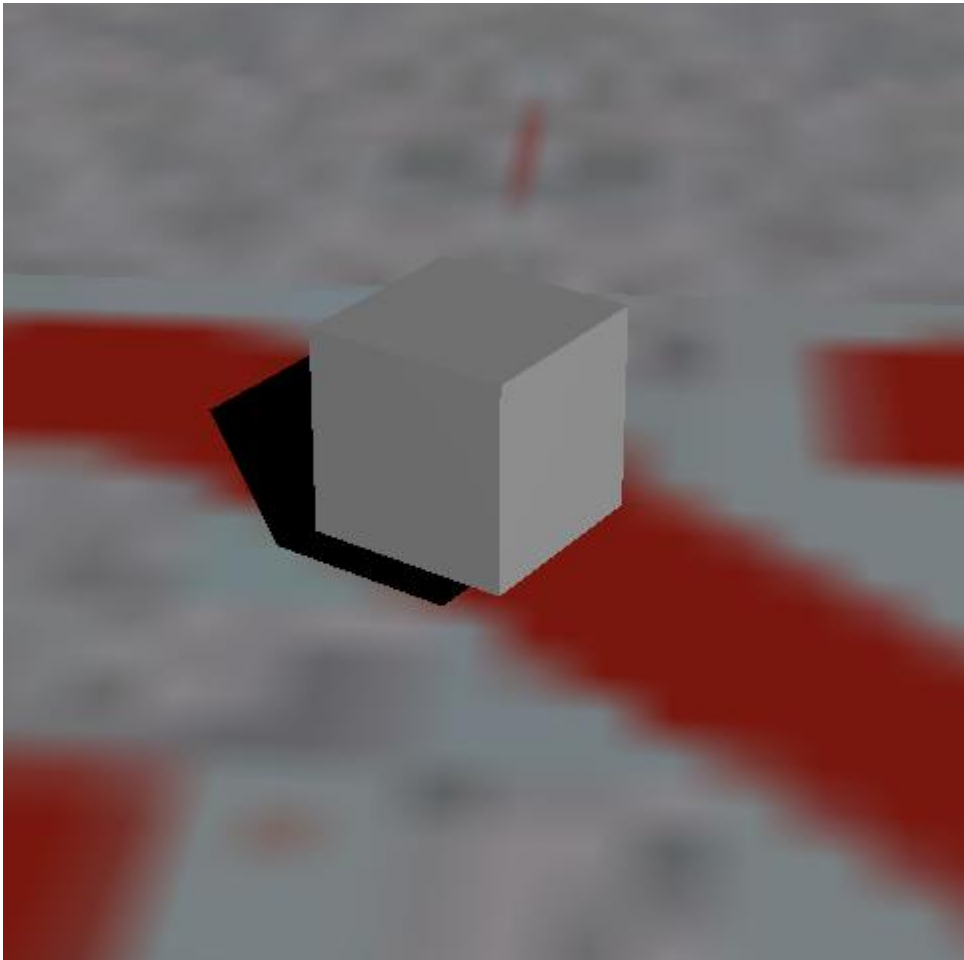
```
BTARGET Brighton Beach
END_MFD

BEGIN_SHIPS
MyShip-01:MyShip
  STATUS Landed Moon
  BASE Brighton Beach:2
  POS -33.4450804 41.1217033
  HEADING 180
END
END_SHIPS
```

Save the scenario file.

We are now ready to run Orbiter and see our *Cube* ship at work.

Start Orbiter and load the *First* scenario we just created. You should see something like this:

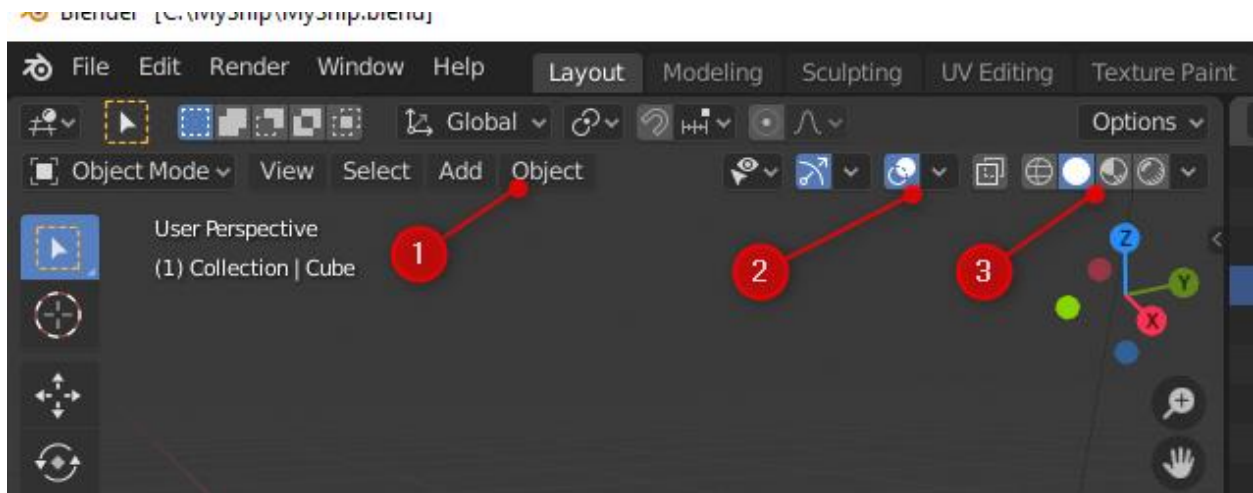


Congratulations! You just built and ran your first Orbiter vessel. We will see if we can do a bit better.

Understanding Normals and Shading

Exit Orbiter and go back into Blender. We are going to edit our Cube object.

A quick discussion of the Blender settings that we will be working with:

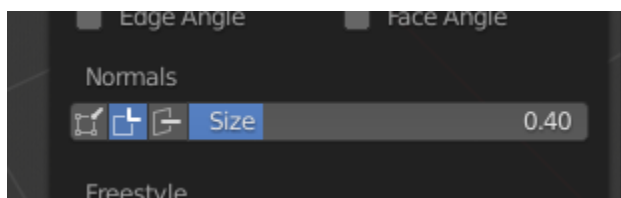


- 1) *Object* menu. This is visible when we are in *Object Mode* (left side of the window).
- 2) *Show Overlays*. We will enable some overlays in Blender that will help us better see what we are doing.
- 3) *Viewport Shading*. Select a shading option to helps us visualize our model while we work on it.

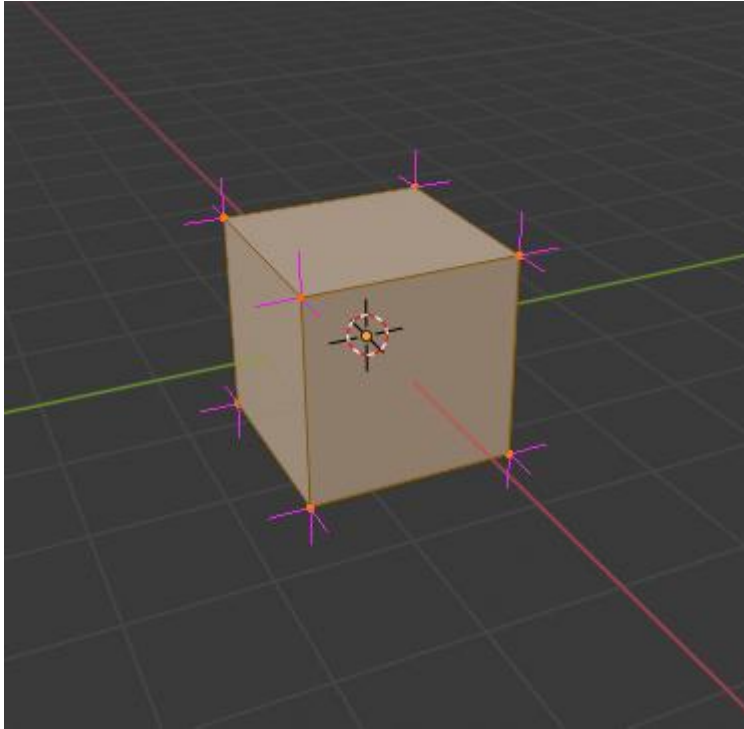
The options on this screen will change depending on if you are in *Object* or *Edit* mode, toggle between them with the *[Tab]* key.

Select the cube (left click) and then press *[TAB]* to go into edit mode.

In Edit Mode, the first thing we want to do is enable *Display Split Normals*. Click on the *Show Overlays* dropdown and enable *Display Split Normals* as shown below. Set the size to about .40 so that we can see them better.



Your cube should look like this:



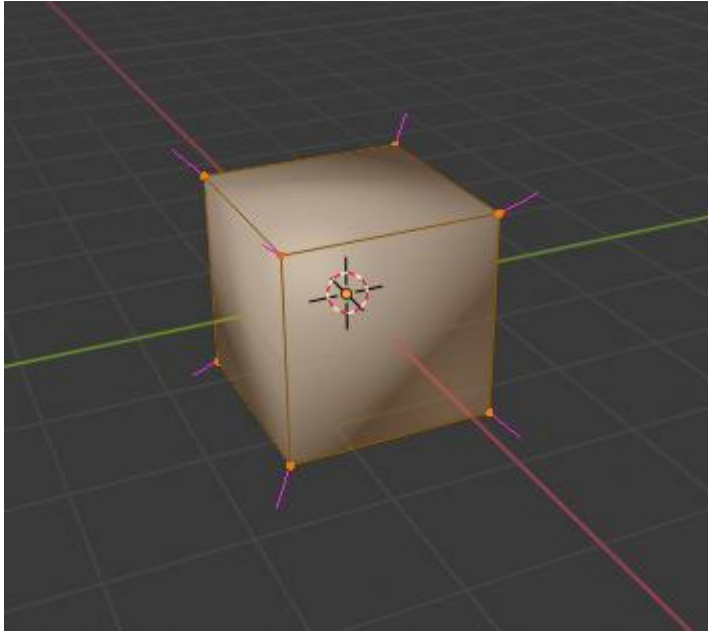
The pink lines represent the *Split Normals* for our model. As you can see, each vertex has three normals associated with it, which is why our corners were sharp when we exported our ship to Orbiter. In Blender an object defaults to 'Shade Flat'. We can change this in the *Object* menu mentioned above.

Toggle *Edit Mode* to *Object Mode* by pressing the *[Tab]* key.

Press the *Object* menu option mentioned above and select *Shade Smooth*.

Toggle back to *Edit Mode* by pressing the *[Tab]* key.

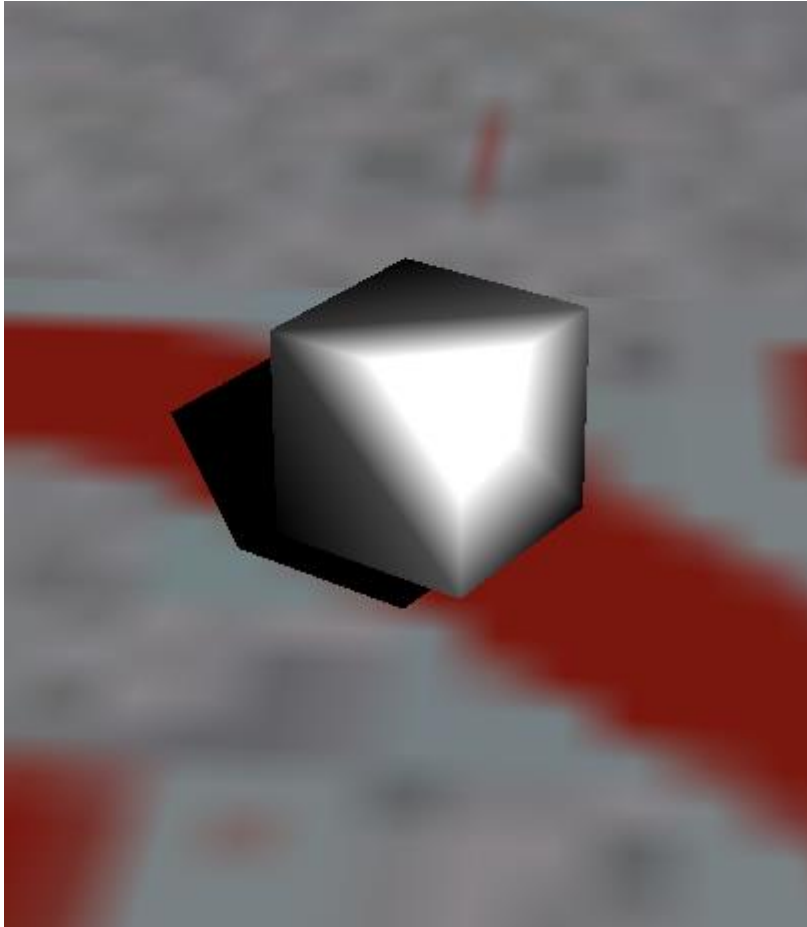
Your model should now look like this:



As you can see, the shading is no longer flat, and each vertex now has a single normal.

Toggle back to *Object Mode* and press *Build Mesh*.

Now, run our Orbiter First scenario again and compare the difference:



This will give some idea of how normal effect the shading of our model. Sometimes you may want a 'faceted' look, but generally our models will have some curved surfaces that help them look more natural, so in general you will want your models set to '*Shade Smooth*'.

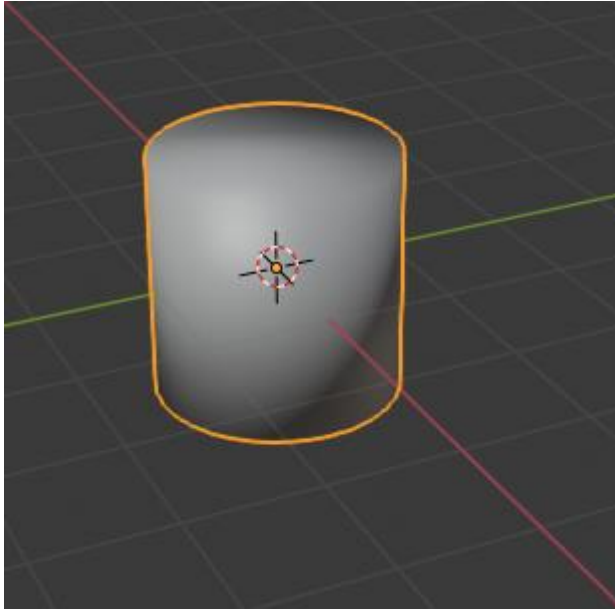
Next, we will look at how to mix smooth and sharp edges. To do this we will use the Cylinder object.

Select the Cube we have been working with and press *[X]* to delete, confirm the delete action.

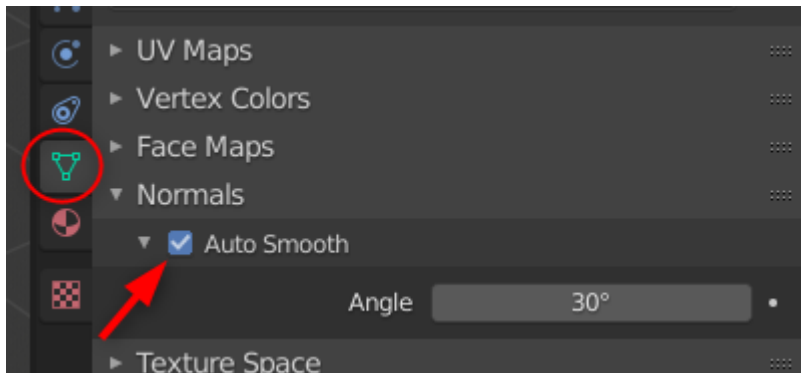
Now press *[shift-A]* to activate the *Add* menu. Select *Mesh -> Cylinder*.

Select *Object -> Shade Smooth*.

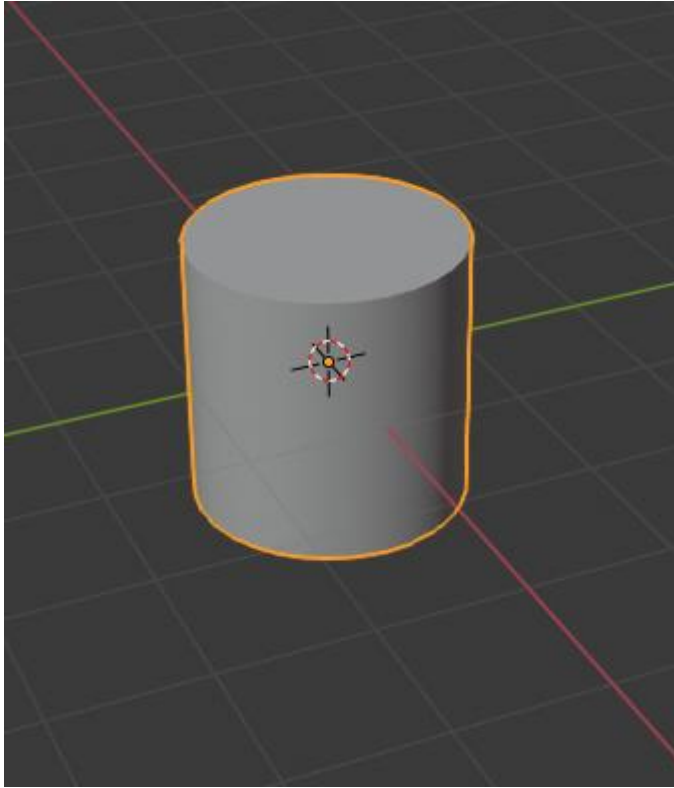
Your cylinder should look something like this:



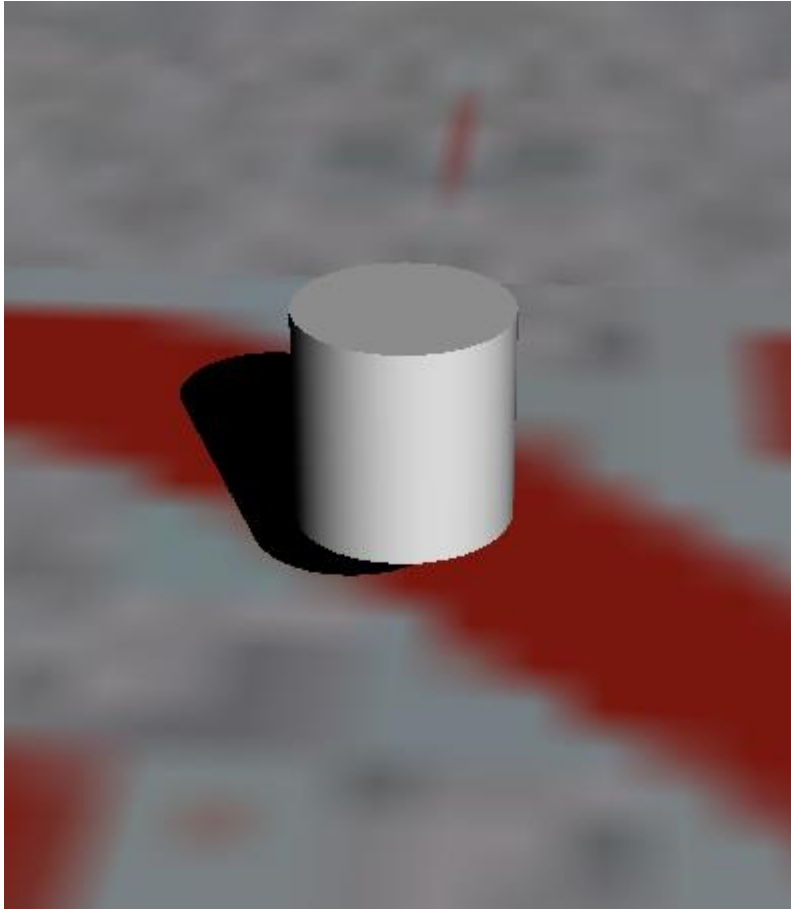
This is a case where we want some of the edges to appear smooth, and others to appear sharp. To accomplish this, we will enable '*Auto Smooth*' for this mesh.



In the properties panel, select the *Object Data* panel, circled in red above. Open *Normals* and enable *Auto Smooth*. Leave the angle at 30. This will smooth edges less than 30 degrees in angle, and leave sharp those that are greater. Your cylinder should now look like this:



Now, *Build Mesh* and run our scenario in Orbiter to see how it looks. It should appear something like this:



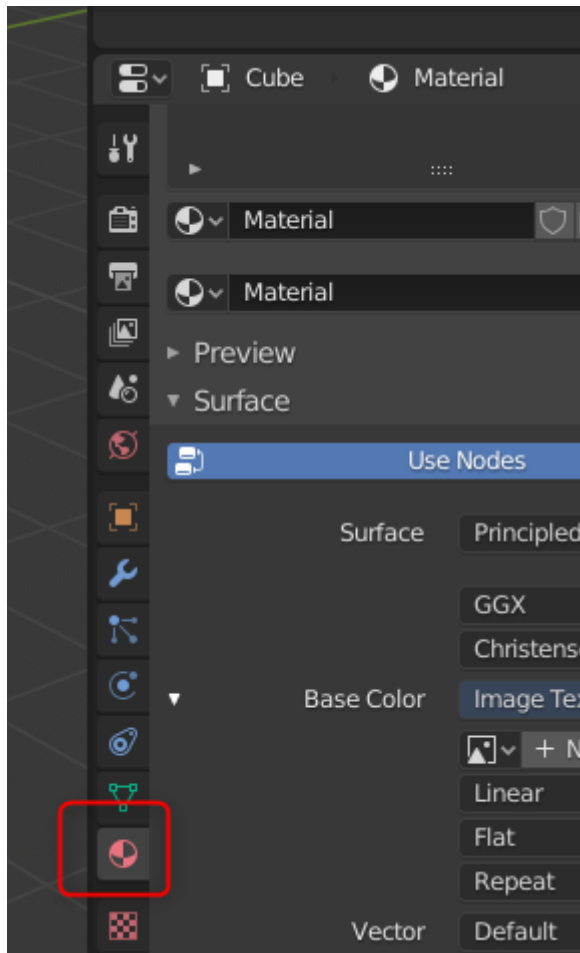
You can adjust the *Auto Smooth* angle to fit the needs of your model. Complex objects can be broken into separate meshes that have their own settings.

Materials

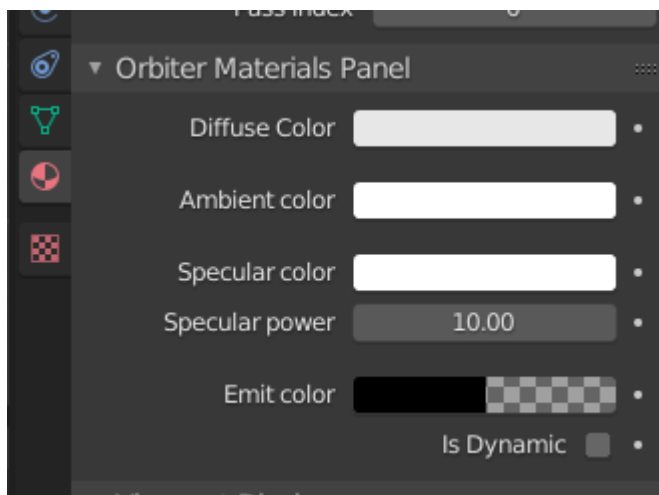
To keep things simple, we will go back to the Cube object to work through adding materials and textures.

With the Cylinder selected, press the *[X]* key to delete. Confirm the deletion. Press *[Shift-A]* to call up the *Add* menu, and select Mesh -> Cube.

Materials and textures will both be applied through the Materials tab on Blender's Properties view:

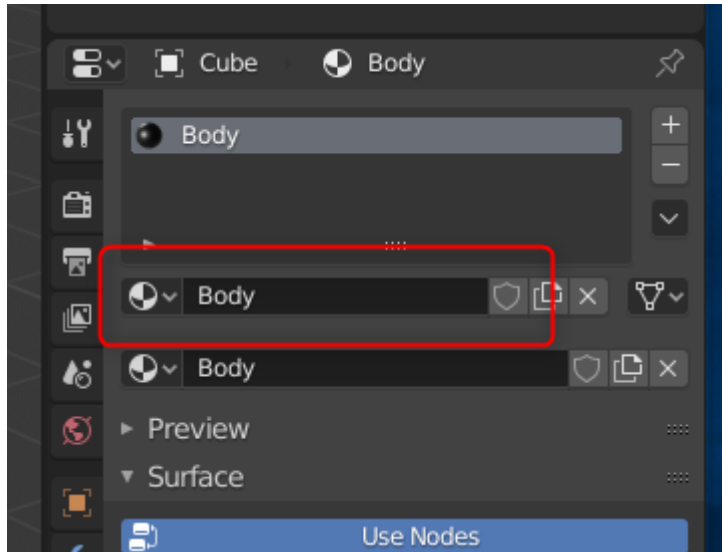


Materials control various properties that effect how the render code interacts with an object. You can create multiple materials in Blender, but the Orbiter Tools plugin will only use the first material for an object. The colors Orbiter uses for a material can be set in the Orbiter Materials Panel at the bottom of the Materials property view:



Other than 'Diffuse Color', you will not see the effect of that color property in Blender, but you will see its effect in Orbiter.

For our sample panel, let's create a 'Body' material with a blue hue. At the top of the Materials property window, find the box circled below and change 'Material' to 'Body': (if *Material* is not already selected, press the dropdown and select *Material* from the list)

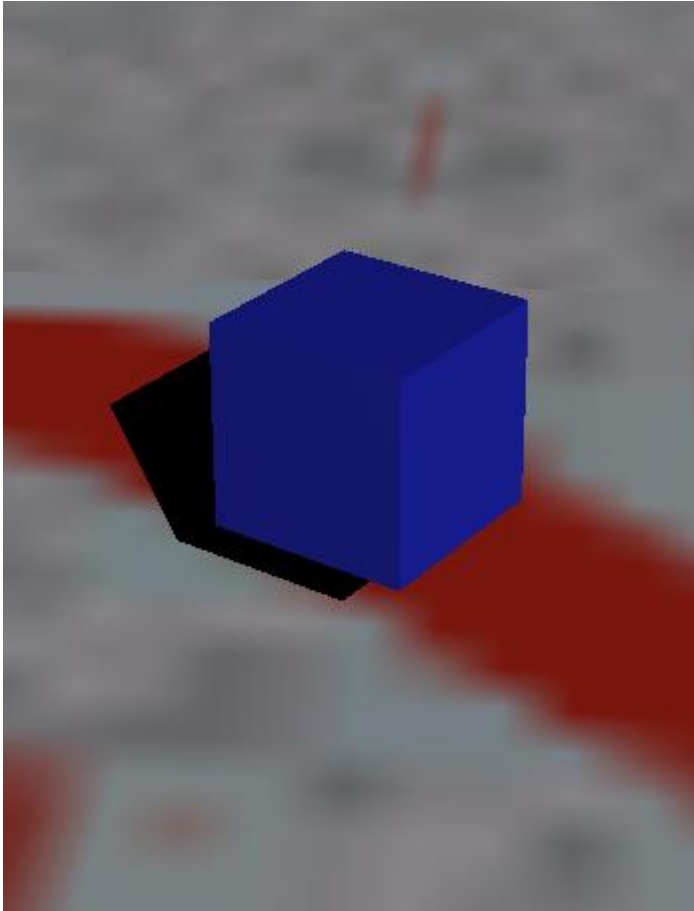


Now, click on 'Diffuse Color' in the Orbiter Materials Panel and select a shade of blue from the color wheel that appears.

You have created a blue material that will be used for the cube object. You can select the same 'Body' material for other objects you create in Blender.

By default, the Mesh Tools plugin will set Diffuse, Specular, and Ambient colors to white (full-on) and Emit to black (full-off). See the Orbiter SDK for a full discussion of these values.

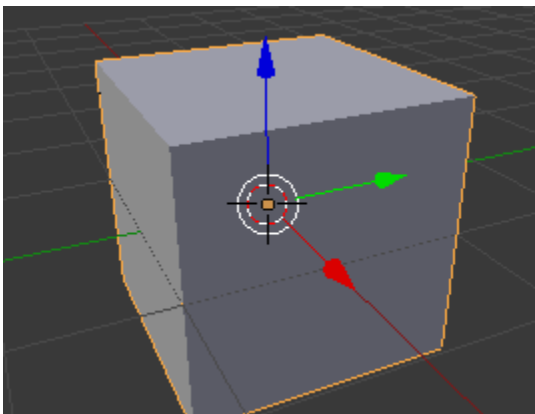
Save, build and run:



Handling Transparency

Now we are going to add a second object that will be similar to a window on our vessel. To keep things simple, we will add it as a *plane* above our cube.

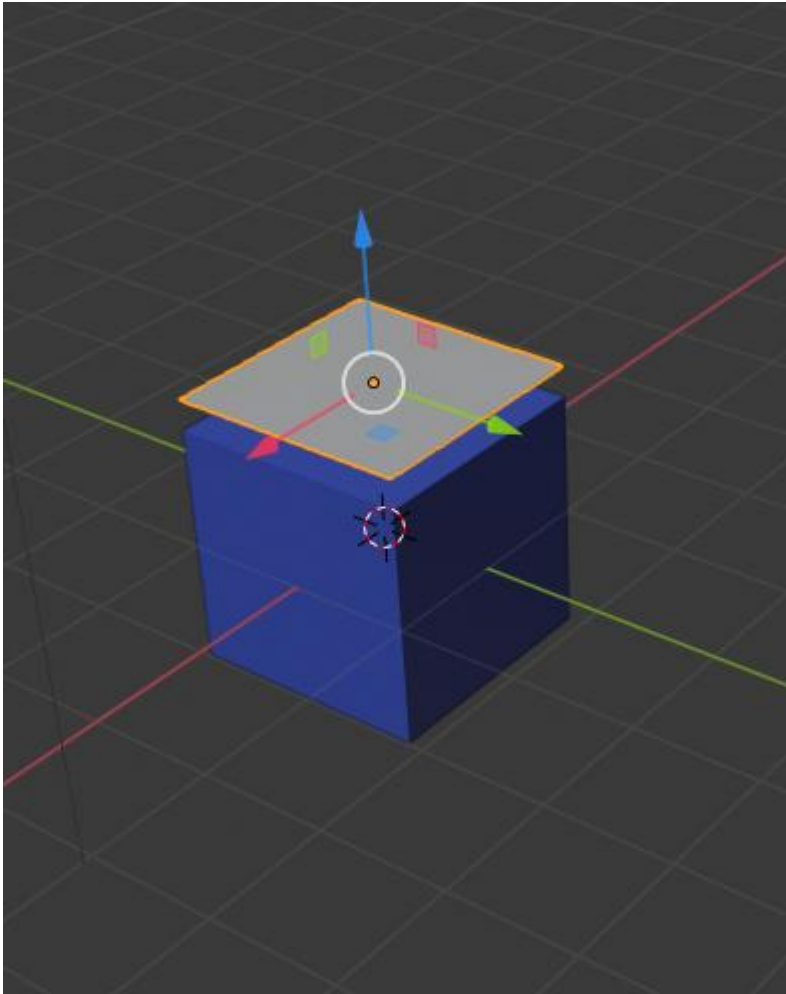
If you have not already, exit Orbiter and go back to Blender. We are going to add a new object to our scene. Blender adds new objects at the location of the 3D cursor, which is the white and red circle in the 3D View.



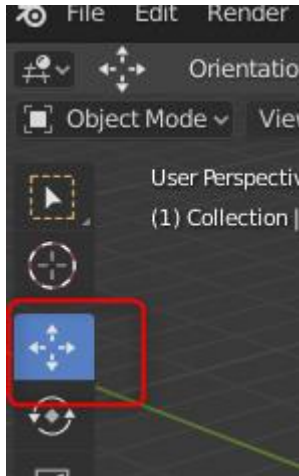
The cursor should be at the center of the *X*, *Y*, *Z* axis. If not, press *[SHIFT-S]* and from the *Snap* menu that appears select *Cursor to World Origin*.

Press *[SHIFT-A]* to add a new object. From the *Add* menu select *Mesh » Plane*. This will put a new Plane object in the middle of our cube.

Click on the blue *Z* axis arrow and drag it up until our plane object is just above our cube.

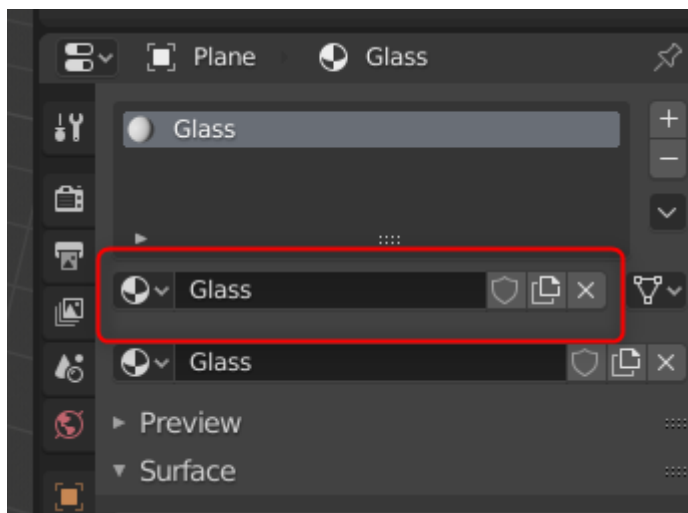


If you don't see the *Move* arrows, you can enable them from the left toolbar:



Now we are going to add a *Glass* material for this object.

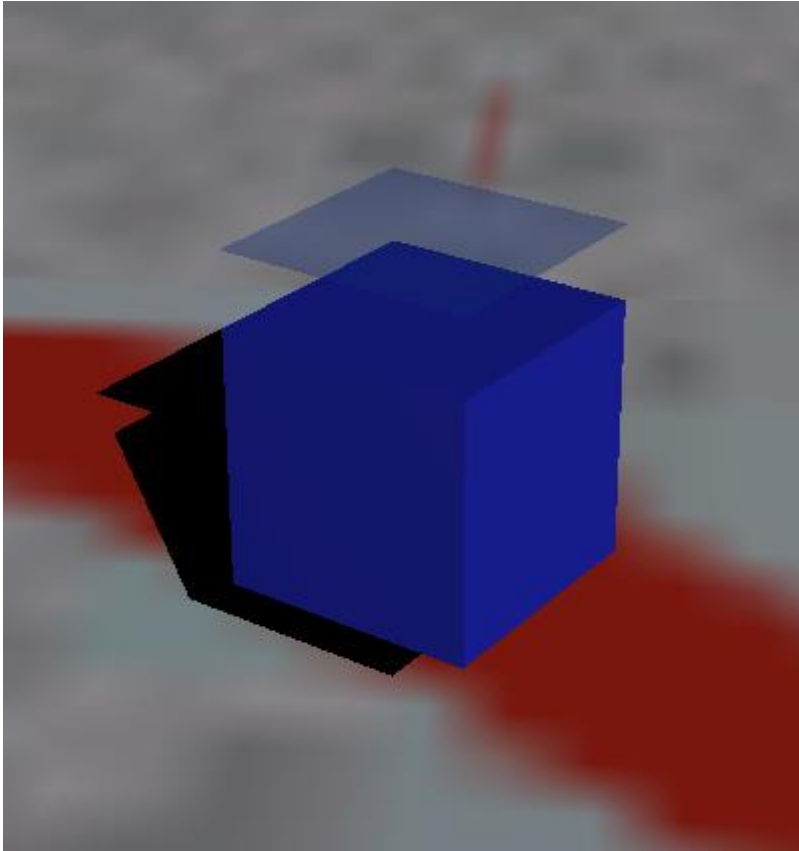
Select the *Materials* property page and click on the *New* button to create a new material. In the material name type *Glass*.



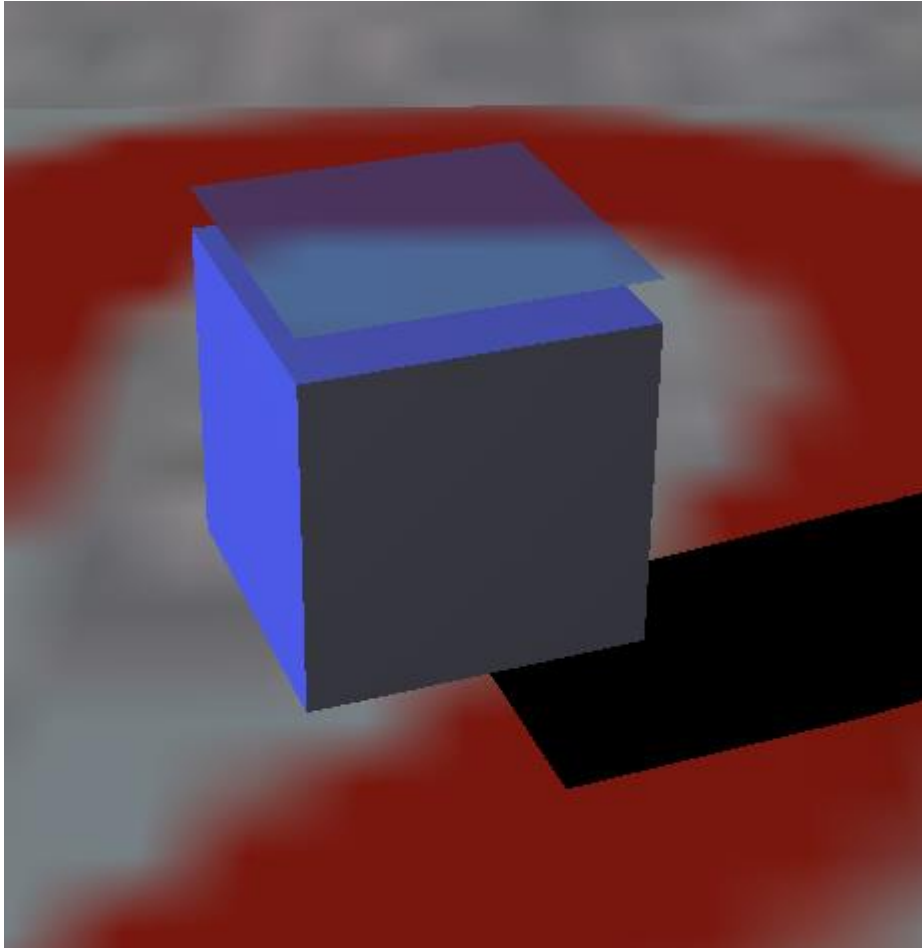
Scroll down to the Orbiter Materials Panel and click on *Diffuse Color*. Select a light blue color on the color wheel, and slide the A slider (alpha) about half way to give the material some transparency.



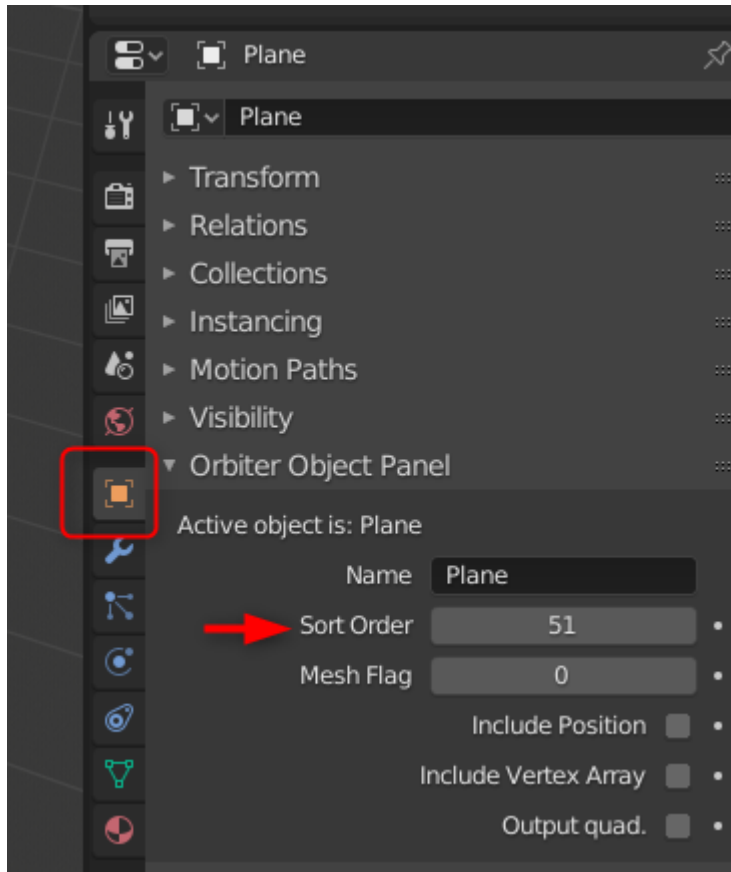
Now Save, Build and Run in Orbiter. You should see something like this:



It's also possible you are seeing something like this:



Here, you can see the landing pad through the cube itself. To get transparency correct, Orbiter expects objects with transparency to render after objects which may be visible through the transparent object. Render order is controlled in the Object properties panel in Blender:



In the *Orbiter Object Panel* you will see the Sort Order control. All objects start with a default sort order of 50. Objects with transparency should have a higher sort order than opaque objects. Set the sort order for our window to 51, build and run in Orbiter. The window should now look correct.

Texturing

Adding a texture to an object gives it visual detail that cannot, or should not be added using modelling or materials. We will add a very simple texture to our cube that will indicate the top, bottom, front, back and sides of our cube vessel. We will create this texture in Gimp. You can also get the texture file from the Git repository where this tutorial also lives.

First let's understand Orbiter and textures. Orbiter requires a texture to be in the .DDS format. Blender understands and can show DDS files, so that is the format we will use when texturing our ship in Blender. The Orbiter Tools plugin does not do any conversion of Blender texture files into DDS, so we must provide the DDS file ourselves. Gimp can do this.

In an Orbiter mesh file, the textures used are listed at the bottom of the file in a 1-based indexed order. Mesh *groups* in the mesh file reference the texture to use by specifying this index. For example, the Delta Glider texture list looks like this:

```

0.375 0.75 0.375 0.400
TEXTURES 10
DG\DGМК4_4.dds
DG\DGМК4_1.dds
DG\DGМК4_2.dds
DG\DGМК4_3.dds
DG\IDPANEL1.dds
DG\DGPILOT1.dds
DG\PSNGR2.dds
DG\PSNGR1.dds
DG\PSNGR3.dds
DG\PSNGR4.dds

```

A mesh group indicates the texture to use by specifying it at the start of the mesh group:

```

MATERIAL 1
TEXTURE 4
GEOM 487 248 ; dgintl_1 0
-0.403454 1.914545 5.163509 0.198101 -0.
0.403453 1.914545 5.163509 -0.296585 -0.
-0.491335 1.465028 6.214284 0.133310 -0.

```

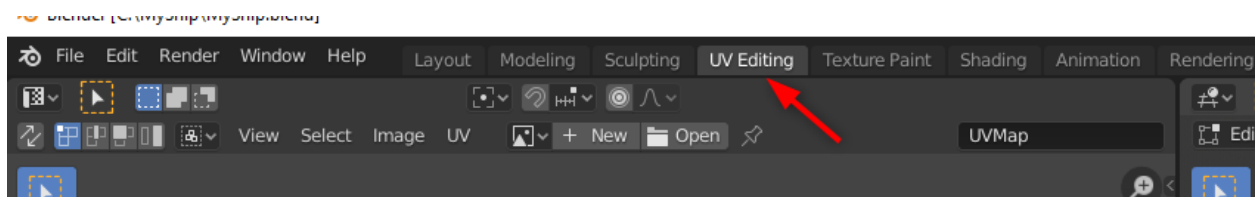
This means use 'DG\DGМК4_3.dds' as the texture for this group. Orbiter will look in the 'Orbiter\Textures\' folder for that file. Note that for the Delta Glider, the textures are in a DG sub-folder, so a file spec of 'DB\DGМК4_3.dds' will be found in \Orbiter\Textures\DB\...

Now we will look at Blender and how Orbiter Tools handles textures.

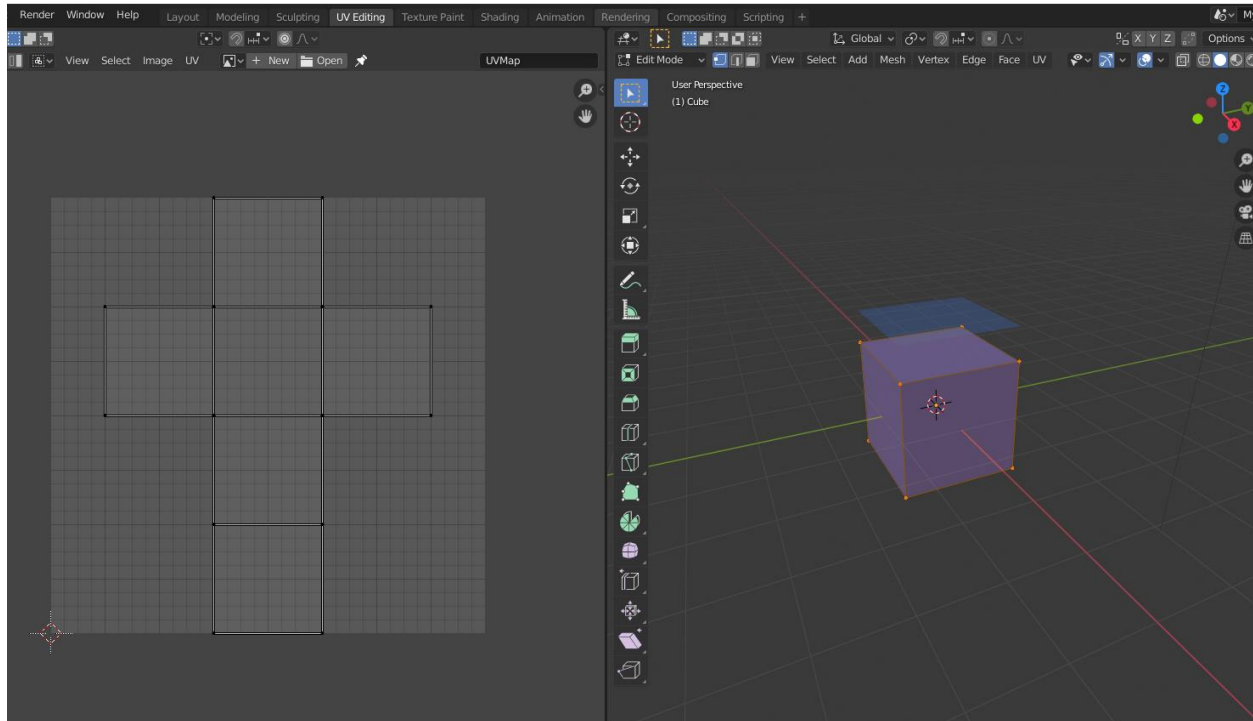
If you are still in Orbiter, exit and return to Blender.

Unlike our model, which is 3D, a texture is only two dimensional, so we will need to 'wrap' the texture around our model. We do that by 'un-wrapping' our model so that the planes of the model can lay flat. This is called UV un-wrapping. For this tutorial we will let Blender unwrap our cube for use. For more complex objects you may need to help Blender by specifying which edges of your model can be treated as *seams*. There are plenty of Blender tutorials that will go into depth on UV unwrapping, but for this tutorial we will keep things simple.

Make sure our Cube model is selected, then from the pre-defined layouts along the top of Blender select UV Layout.



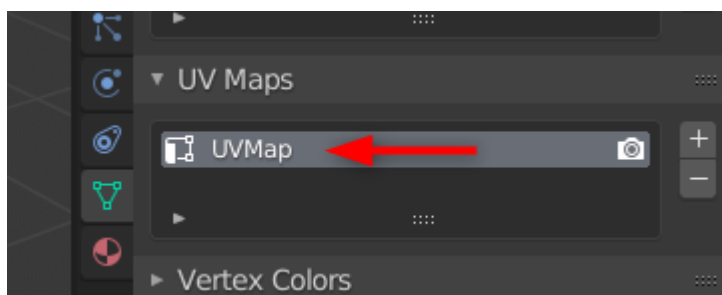
The screen will split and on the right side you will see our model, and on the left you will see Blender's layout of our UV map.



You can export the UV image in the left pane to assist you in creating your texture file. You can also use Texture Paint to mock up, or create a rough guide for how you want your final texture to appear. Note that the image you create in Blender is not usable in Orbiter, but can be useful in another image editor as a guide.

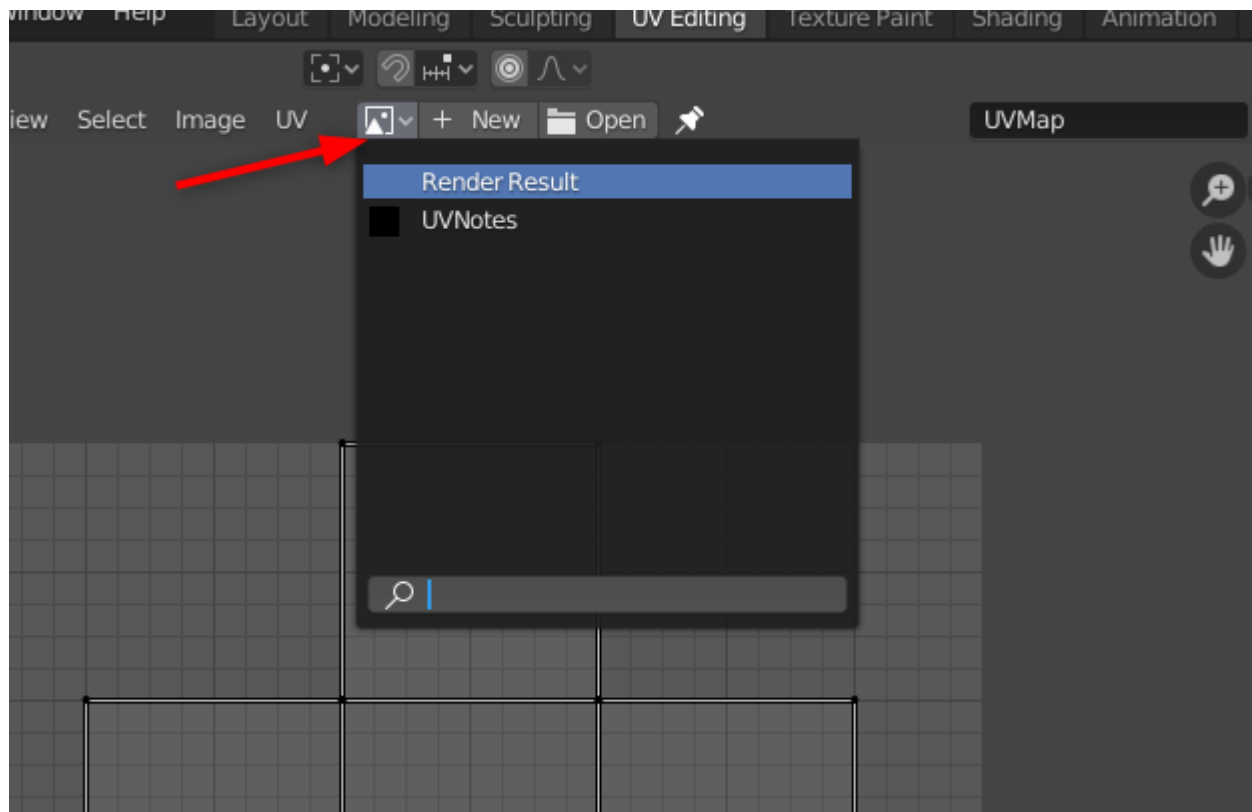
That process is another tutorial.

For now, there is nothing else to set on this screen. A few things to note however. On the top left of the UV pane (left side) you see the name of this layout *UVMap*. If you open the *Object Data* properties for the cube, you will see the list of UV Maps for this object:



You can have multiple maps for an object. Orbiter Tools will only use the first map on this list when creating the mesh file, so make sure you either have only one UV map, or make sure the correct map is the first one on this list.

Also, at the top of the UV pane there is a dropdown that will let you select an image to show along with this UV map.

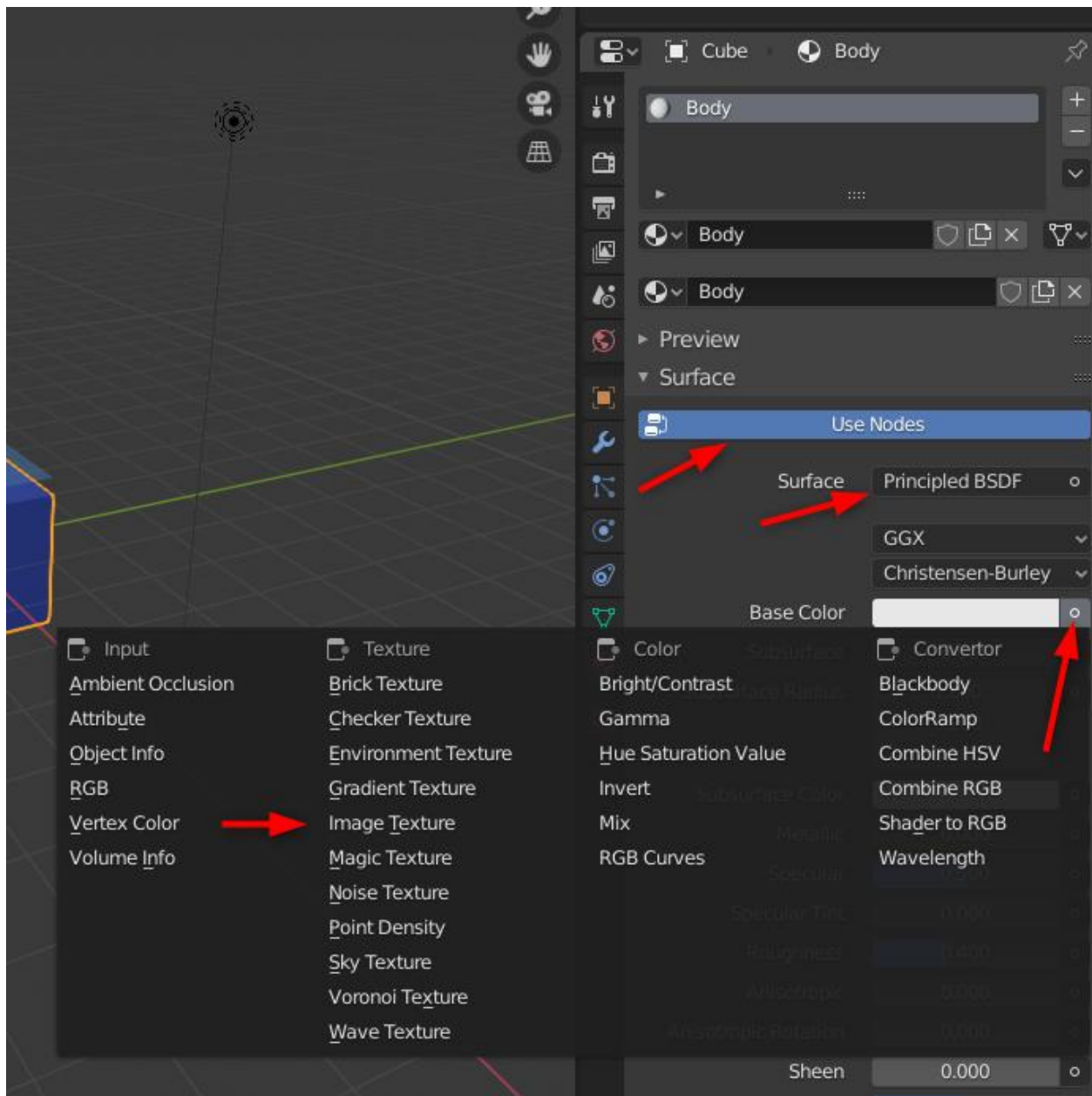


This is convenient for when you want to use Texture Paint, or view some image along with the UV maps. It is important to understand this is NOT the image that Orbiter Tools will use when creating the mesh file. That image is set through a node, and we will do that next.

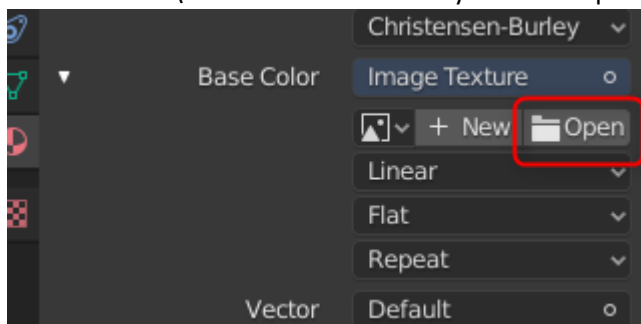
To specify the texture to use for this mesh, we are going to add an *Image Texture* node to our *Body* material. We can do this in the *Material* properties panel directly without worrying about the Node editor, this will make things easier for now.

Open the *Layout* Blender configuration (select *Layout* from the top bar).

Open the 'Materials' property pane in Blender. Our 'Body' material should be selected. In Blender 2.81, textures are managed through Nodes. Fortunately, we can get what we need without using the full nodes editor. In the Materials panel, make sure *Use Nodes* is selected (it will have a light blue background). For Surface select *Principled BSDF*. Now, next to Base Color click the little white circle. See below:

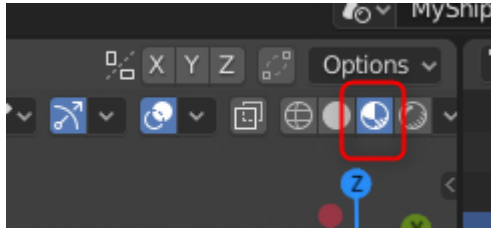


This will tell Blender to expect a texture as the base color. Select *Open* below Base Color, and navigate to the Orbiter\Textures folder where you have copied the BodyTex.dds file.

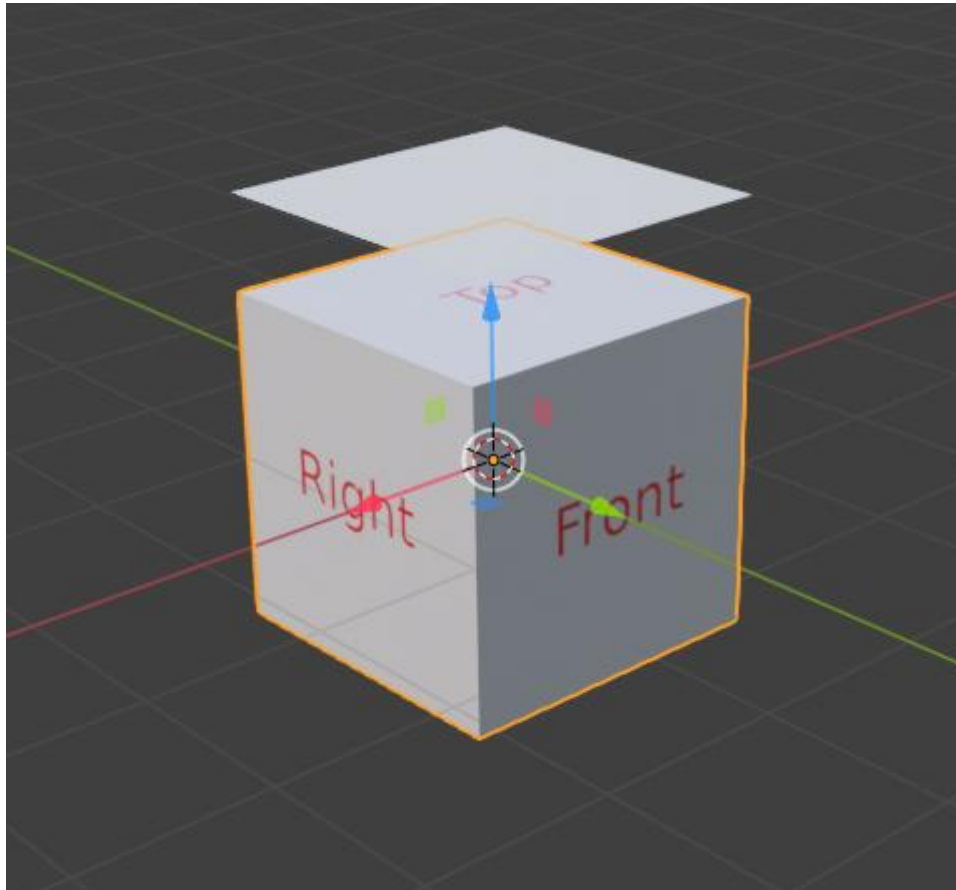


Also, if your diffuse color is still set to non-white (blue is what we set) that will show through your texture, so either set diffuse back to white, or set it alpha to 0.

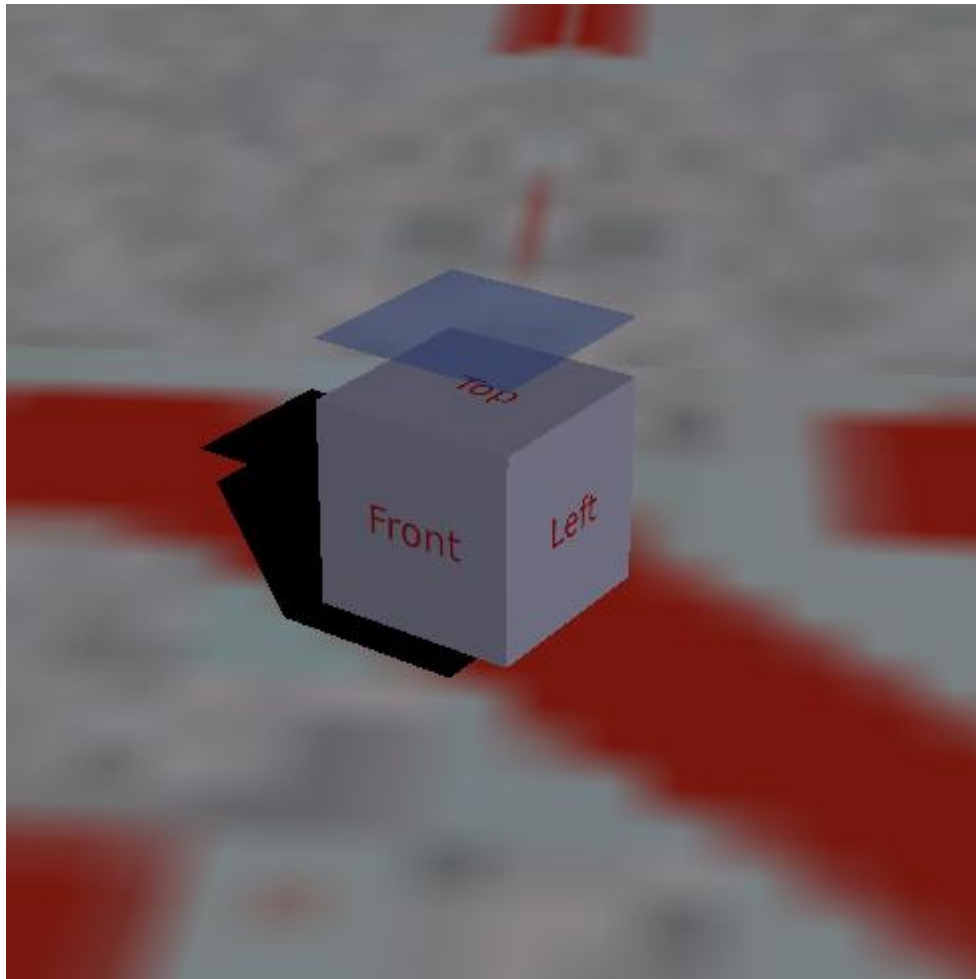
In the 3d Viewport on the right, select the viewport shading option (upper right) as shown:



You should now see a textured view of your vessel:



Finally, *Build Mesh* and run in Orbiter:



Congratulations, you have now created a textured Orbiter vessel.