

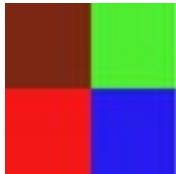
Workflow between Maya, Photoshop and Unreal Editor - Static and Animated Meshes.

This tutorial is meant to demonstrate the workflow only and is not a guide on how to import complex models into Unreal.

Open up Maya and set your grid (display / grid / option box) to size 128 with grid lines every 16. 1 subdivision should do it. This is so that 1 grid unit inside Maya is the same as 1 grid unit in Unreal so you don't have to scale your object inside Unreal.

Click create / polygons / cube. Open up the options of that selection (the box next to the name) and in size type in 32, 32, 32.
Minimize Maya for the moment and open up Photoshop, gimp or whatever your image creation program of choice is.

Open up Photoshop and create a texture that is 64 * 64 with a white background. Make sure it's RGB.
Divide up your image into 4 sections and fill each section with a different color. This is an example:



Save your file as a tga file 32 bits. If we save it 16bits, Unreal will make turn it into a greyscale image.

Minimize Photoshop.

Go back to Maya and open up your Hypershader (window / rendering editors / hypershader)

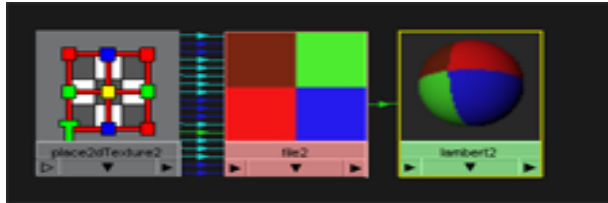
In the hypershader create a new lambert texture by middle mousebutton drag it down to the work area (or just click it!):

Open up the attributes for this material by double clicking on it.

Click the box next to the color node.
And select file... Click the browse button and browse to where you saved your image file.

Select it and click ok.

You should now have something like this in your hypershader:



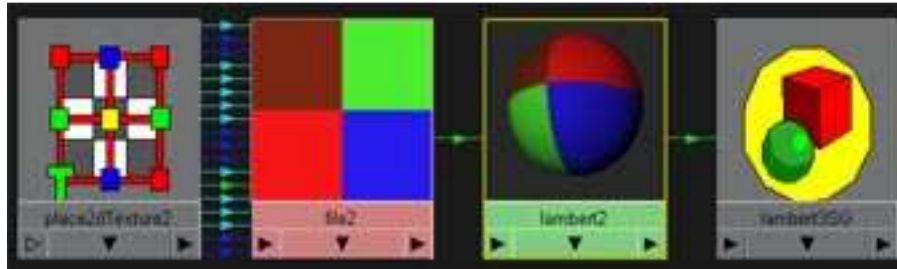
We will add one more thing, if you are working on a mesh that will be animated:

Select your lambert material and hit the button that shows you the input and

output streams:



You will see this:



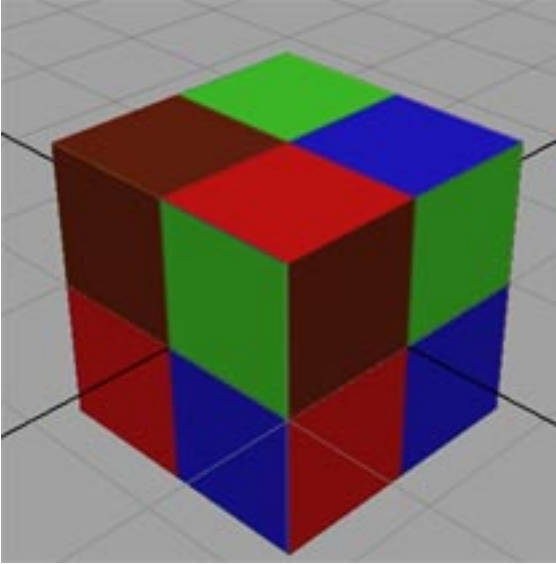
Double click the node (those things you are looking at are called nodes, btw) called "lambert 3SG" and open up the attribute editor. Click the tab called "lambert 3SG" and find where it says "shading engine". In the box next to it, type in a name here instead. Preferably something descriptive such as "DoomBox" or something that makes sense.

This will be helpful later on if you are texturing a mesh that will later on be animated.

Let's go ahead and triangulate your mesh. Select your Box of Doom and go Polygons / Triangulate. You will now see that your mesh becomes split in the middle on all sides. The edges form a triangle instead of a square.

Apply your texture to your polygon cube by middle mouse dragging the material on top of your cube. Hit the number 6 on your keyboard to go into hardware texture mode so you can see your textures.

You should have something like this now:



Now for the interesting part. UV Mapping this box! To make a long explanation short, UV Mapping is basically how the texture is aligned on your object.

We want each side of the box to have its own color, so here's how we go about doing this:

Click: Panels - Saved Layouts – Persp/UV Texture Editor.

The interface for Maya changes slightly. Don't panic. On your left side is your regular Perspective viewport. To the right we will do the UV Mapping.

Ok. Let's start.

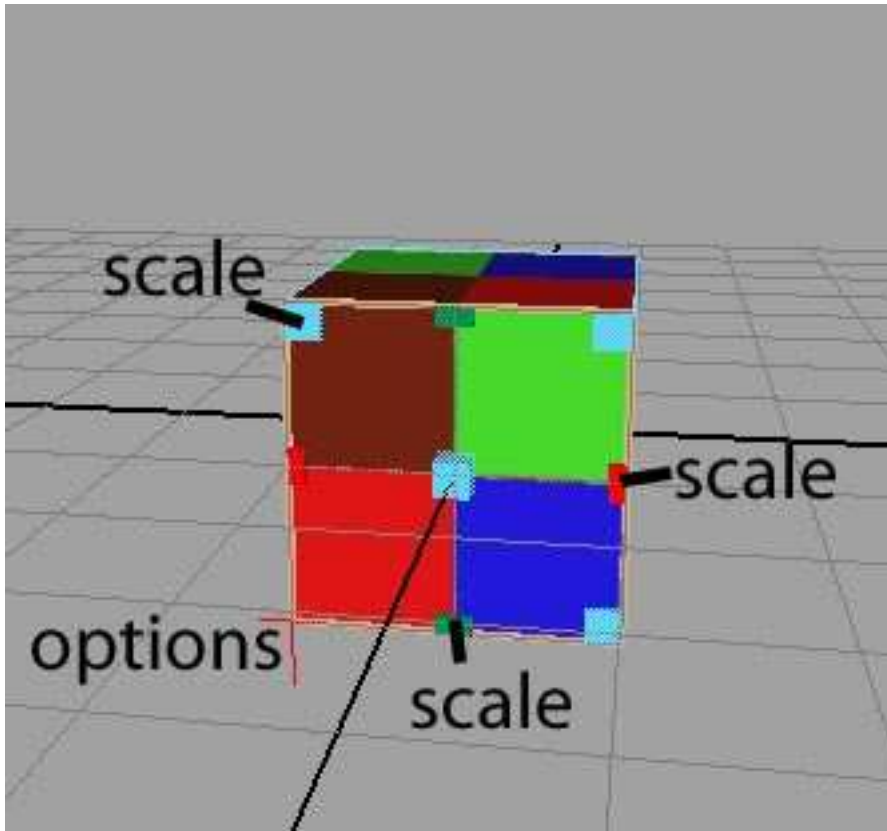
Right click and hold down your mouse button over your cube of doom and select "face" from the popup menu.

Select two of the faces making up one side. You'll now see our texture show up in the other viewport... but you can't manipulate anything yet. We need to assign a UV Map to this face before we can do anything.

With your faces of choice selected, click Edit Polygons / Texture / Planar Mapping and hit the options box.

An array of options is here for you to experiment with. But since our object is a cube we can get away with having "Smart Fit" selected, clicking "Fit to best plane" and hitting "Apply"

You will now see that you have a manipulator where you can scale and move the texture around. See image:



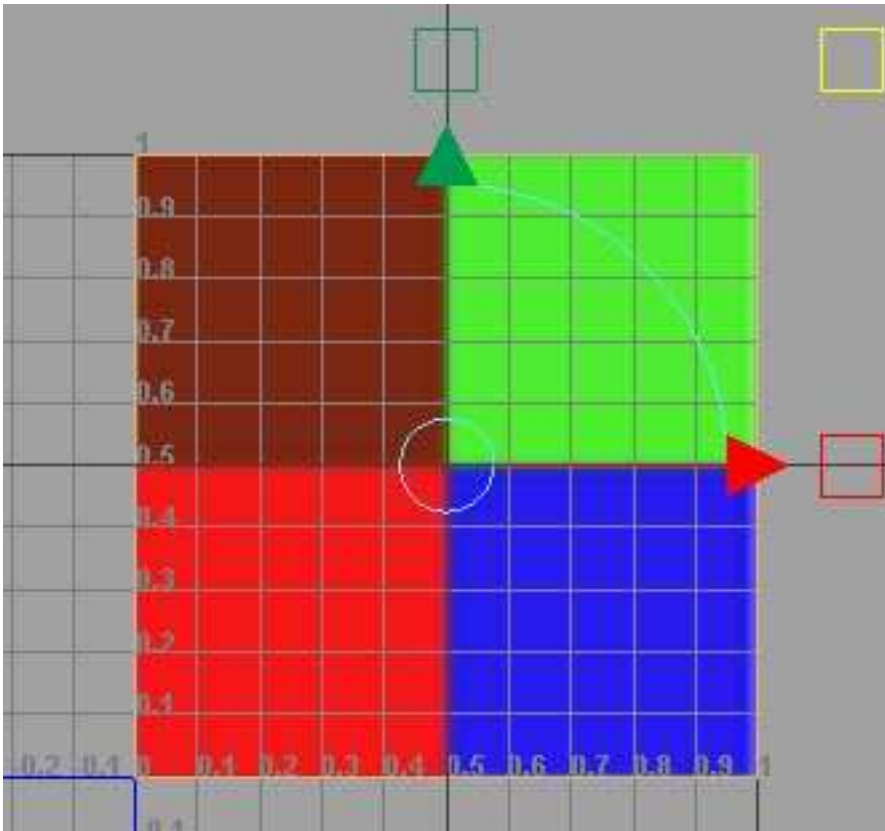
The light blue box will scale your texture equally on all axis. The red and green box will scale your texture only on the axis that they are on. Y Axis for green, X Axis for red.

If you click the option handlebar (red upside down L) you can then rotate the texture however you see fit.

If you glance up at your channelbox you can see exact numbers for what you are doing graphically inside your perspective viewport.

Ok! Now we are getting somewhere. We have a texture, and we have a way to manipulate said texture.

Now head over to your UV Texture Editor and let's take a look at what's going on there.



Again; don't panic. What you see here is just a different way to do the exact same thing we did in the Perspective viewport. If you grab the red box you scale your texture on the x axis, green for y and the yellow for both axis's.

Since we have a cube it would be easy to scale this thing down (the UV's) down to one color and just move it over, but since we are aiming for more difficult tasks such as creating a character for Unreal, let's go over how to manipulate each and every UV point.

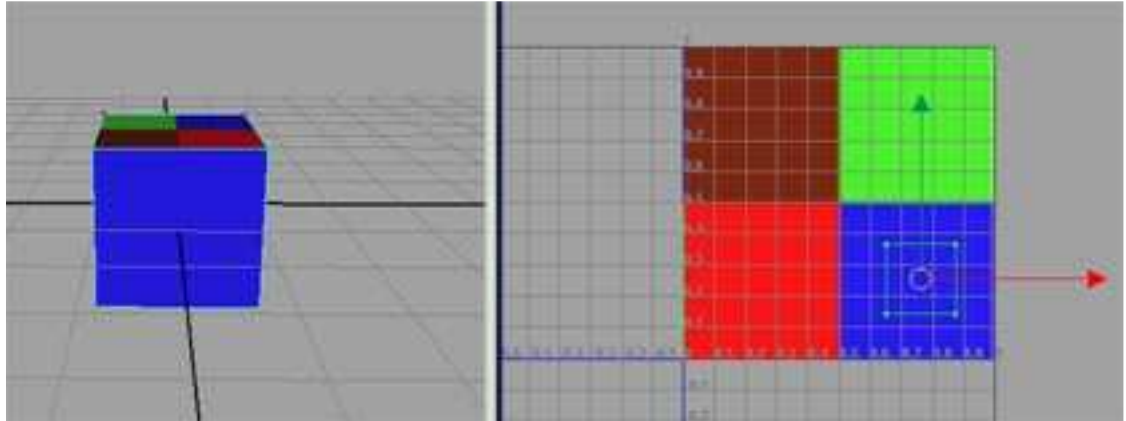
With your face still selected right click and hold inside the UV Tex Ed and select UV. Nothing happened, right? Wrong.

If you marquee select the whole texture you will see that you can now select the UV points and move them, scale them, rotate them and so forth. You can even move each UV point individually which will be very, very helpful when you get into texturing more complex objects.

If you would like to go back to the easy to use handles that we had above go to your channelbox and click the word "PolyPlanerProj1" and then hit the "T" on your keyboard (with your face still selected). You can now manipulate the texture this way again. If this doesn't work, make sure Maya has Construction History enabled.

Now, move your UV points of this face so that in your perspective viewport you see that the face you have selected is covered in just one color.

You should end up with something like this:

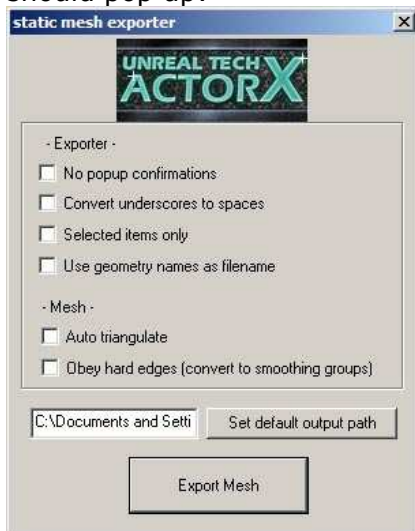


Now continue around the cube until all faces only have one color. When you are done, let's export this masterpiece into Unreal!

Ok. You should now have a polygon cube that has 1 64*64 texture where each side of your mesh has just one color.

Delete history on your static mesh. No need to bother with that now that your Cube is ready for exporting.

Go down to your command line and type in "axmesh;". A window like this should pop up:



If it doesn't, then make sure the Actor X plug in is loaded. You can check this by going to Window / Settings and Preferences / Plug In Manager. ActorXtool.mll should have checkmarks next to it.

First off, set your default output path. This is where Actor X will save your file. If you have not triangulated your mesh yet, go ahead and click the Auto Triangulate. All meshes have to be triangulated inside Unreal or holes will appear in your mesh. Also check "Selected Items Only" just in case you have more than one mesh going on inside Maya. Ignore the rest of the options for now.

Hit Export Mesh and if everything went well, you will get a message saying that 1 part, 12 polygons and 1 materials were exported. Success! You will now have an .ase file on your desktop or where ever you set your default output path.

Now open up Unreal Editor.

Go to the texture editor and import our texture (if it's not open: View / Show Texture Browser) (File / Import) that we applied to our cube.

Make sure that the name you give it is the same name we used inside Maya (the filename). So if the filename of the texture is "MyTexture" then the name of the file inside Unreal Editor should be the same.

Now go to your static mesh browser and import your cube of doom. Give it a new Package, group and name. Doesn't matter what you call it as long as it is descriptive to you and is not the same name as anything else already existing.

Import it and thanks to the fact that we named the texture the same thing in here as in Maya, the texture is already on the cube with the UV mapping correct.

Addendum: If you want to change your texture name or type, you can open up your .ase file with your favorite text editor and look for this:

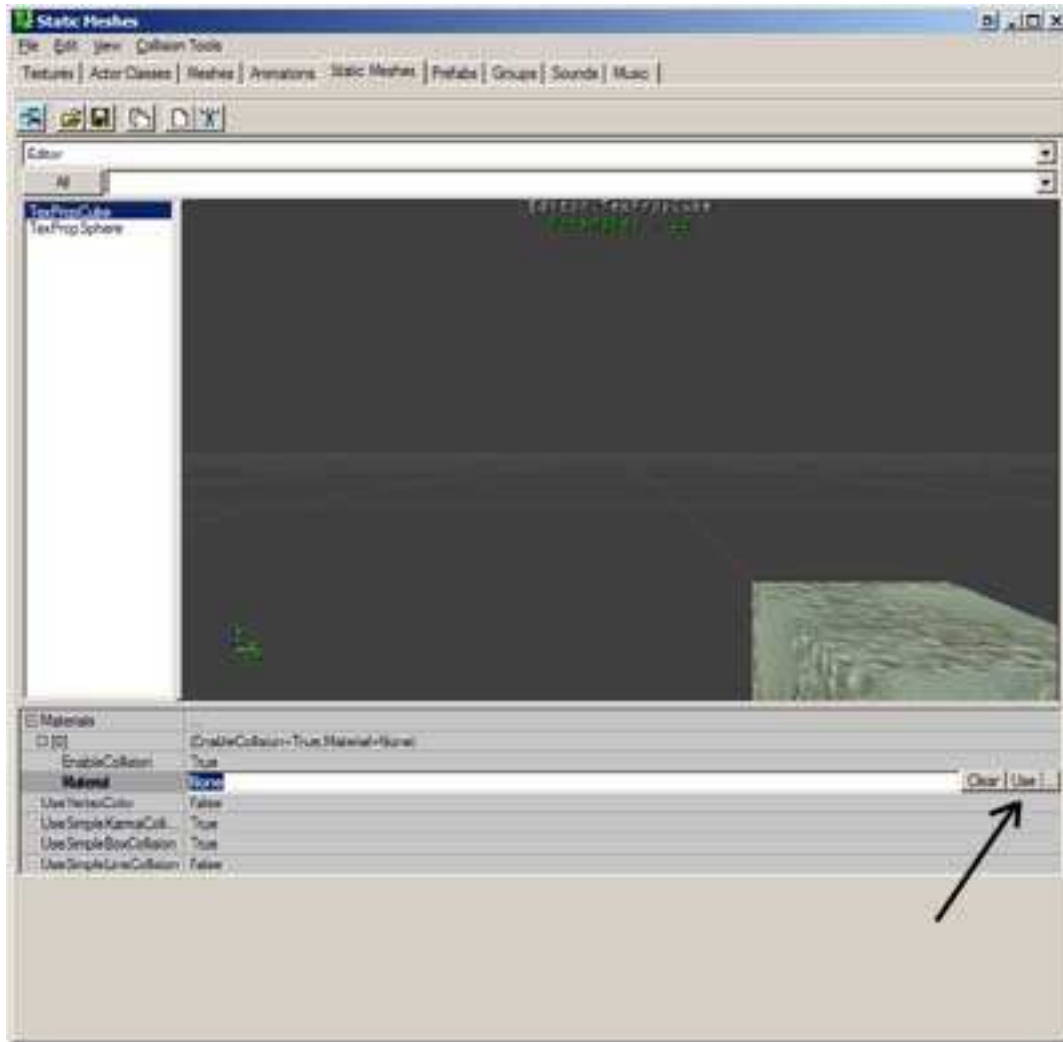
```
*MATERIAL_NAME "lambert3SG"  
  *MATERIAL_CLASS "Standard"  
    *MAP_DIFFUSE {  
      *MAP_CLASS "Bitmap"  
      *BITMAP "C:\color_map.tga"
```

Under the bitmap section you can change your file name to whatever you want. As long as you import the texture with that name Unreal Editor will automatically find which texture belongs to which mesh.

So for example... I want to rename my texture to "MyTextureCube.tga". All I have to do is go into the .ase file, change the name, and when I import it into unreal editor give it that exact name and the editor will match it up with the cube we made in Maya.

What? You ignored naming things? Well, alright. What you do is go to your texture browser, click the texture you want the box to use. Then click the static mesh tab, and click Materials, expand 0 and click Material. Then hit Use. This uses the currently selected material in the texture browser as the texture.

Be sure to save both your texture package and your static mesh package.



Note that for this workflow to function properly, each name **MUST** be unique. If you have two textures of the same name, Unreal Ed defaults to the texture that was last imported.

You can now open your favorite unreal map and place your box of doom in the map.

Animated meshes.

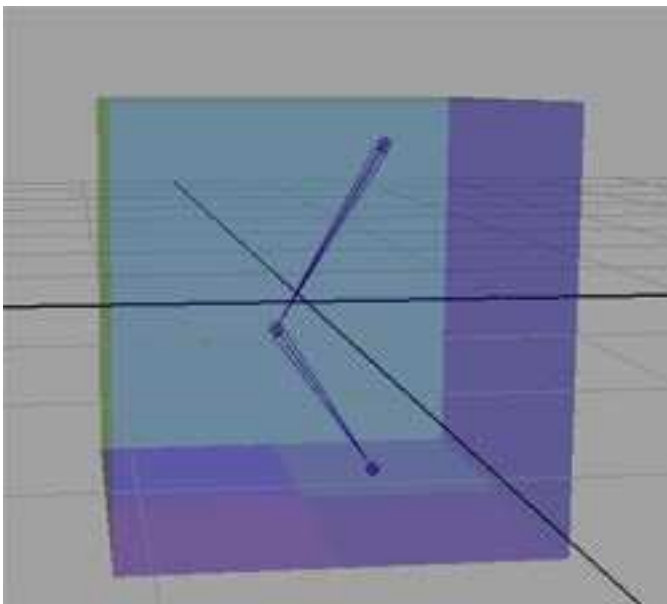
So you are working on a mesh that will be animated, eh? Ok. You will do almost exactly the same as the static mesh, but your workflow will differ just a little bit at the end. You have it a little easier than the static mesh people since you will be getting your models already textured.

So why should you care about the textures? Well... static meshes and animated meshes are two completely different beasts but you still have to import them into Unreal and yes, that includes the textures.

So what's different?

Let's use the same box we used in the above example: Le box de la doom.

Add a bone into it and bind it with smooth skin (unreal only handles smooth skin binding). Now this is for example only, so a simple bone structure and default skinning will do fine. Something like this will be great:



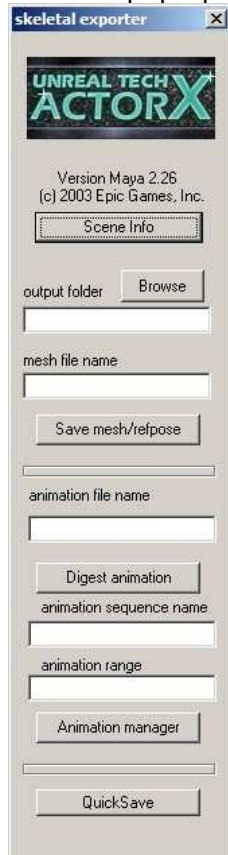
You ask, how do I do this? Well, it's easy as cake, dear! In Maya, go to the side view by hitting the spacebar once, clicking in the side viewport and hitting the spacebar again. Now either hit F2 or click the drop down menu in the top left part of Maya. Select Animation.

Under the Skeleton menu, select joint tool and begin building your skeleton. Build from top to bottom until you have roughly what I have above here. Next select your box and the skeleton, go up to Skin and then Smooth Skin. Click it. Go ahead. Your object of doom now has a skeleton and is skinned to this skeleton. This means that if you move the skeleton, the skin will move with it.

When you start getting into Unreal modding, you will probably use Unreal's default skeletons and bind your mesh to those instead.

Now, go down to your command line and type in "axmain;" without the quotes.

This will pop up:



Yes, this is where we export animations also, but let's concentrate on the textures, shall we?

First off, set your output folder (could be your desktop for this example). Now, name your mesh something descriptive. I called mine BoxDoom, now hit "Save Mesh".

You will find that you have two files on your desktop now. BoxDoom.PSK and X_ModelInfo_BoxDoom.LOG. It's the log file that we are interested in.

Open it up and find this section at the top:

```
= materials =  
* Index: [ 0] name: DoomBox  
  Original bitmap: color_map.tga
```

This is where Unreal Editor looks to see where the texture of your animated mesh is, in particular the "name: DoomBox" section.

You can close it for now.

Now open up Unreal Editor and let's import this box.

Go to your texture editor and import your texture first. Mine was called "color_map.tga" as you can see, but when you import your texture, you need to name it the same as the Index name. DoomBox. Let's do that.

Next go to your animation tab and import a new mesh. Call it anything, but be sure to check "Use Maya Coordinates" since Y is not up in the Unreal World. You can now see that because we named our textures correctly, we don't have to hunt through our texture packages to assign them to our meshes but unreal does it for us.

If you ever get confused about your naming scheme (which you shouldn't) you can always open up the X_ModelInfo*.LOG. file and check what you or the texture artist named it.

Of course... you can always ignore the naming schemes (although I don't recommend it) and assign the textures to your models inside the Unreal Editor.

Thanks for reading. I hope this tutorial was helpful in showing how to move between Maya and Unreal to create new meshes and objects.