

Halo Editing Kit

Compiled Tutorials by SuperSunny (Sunny Sharma)

Only for Intermediate or Advanced HEK Users. Beginners might be a bit confused.

References

Chapter 1: Textures and Shaders

Chapter 2: Simple Box Modeling Setup.

Chapter 3: Glow

Chapter 1: Textures

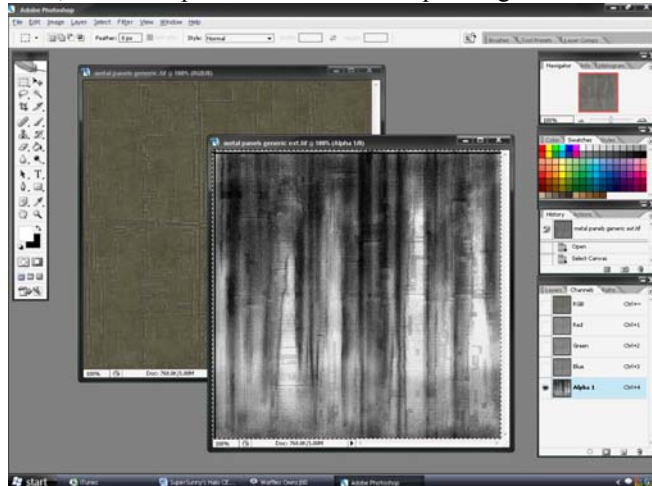
Part 1: Beautify Textures in Photoshop

Start by opening up

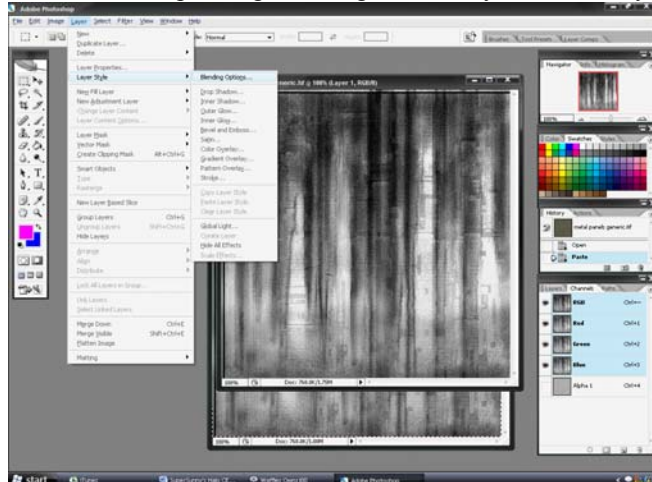
levels\test\b30\metal panels generic ext.tif

levels\test\b40\metal panels generic.tif

Now, take the alpha channel of “metal panels generic ext” and paste it into the normal RGB Channel of “metal panels generic.”



Now, in “metal panels generic,” go to the Layers menu, Layer Style, and then Blending Options

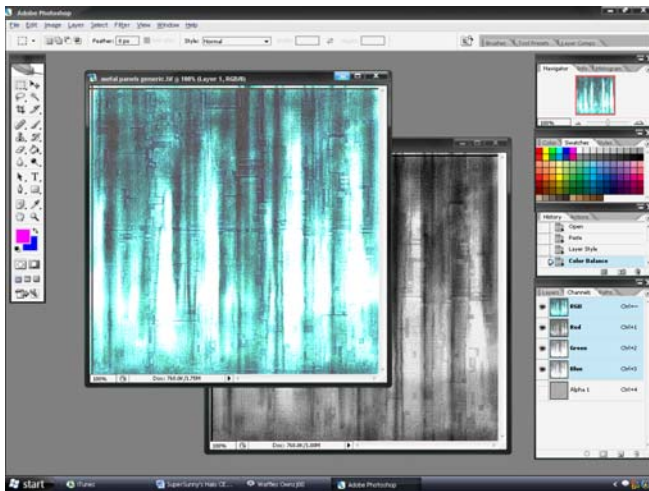


Change the “Blend Mode” to “Color Dodge.”

Click Ok.

Go to Image, Adjustments, and then Color Balance.

Play Around! Get any interesting Color Effect.



Click Ok.

Now, take the Red Channel, copy the image in it, and paste it into the Alpha Channel. Save as a tif. Make you sure you check the box that says “Discard Layers and Save a Copy.” Exit out of both the images, and open up the file you saved again. If it tells you to save or don’t save, just select don’t save. We had to exit so the layers become one again.

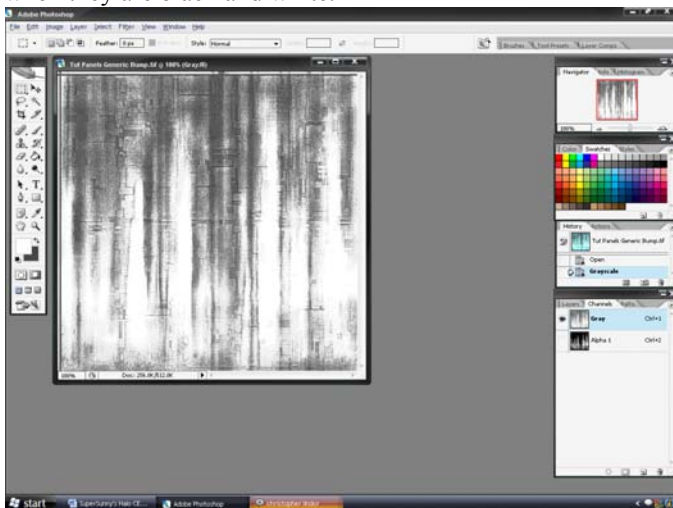
Part 2: Creating Bump-Maps out of the texture

Time for some fun! Bump-Maps!

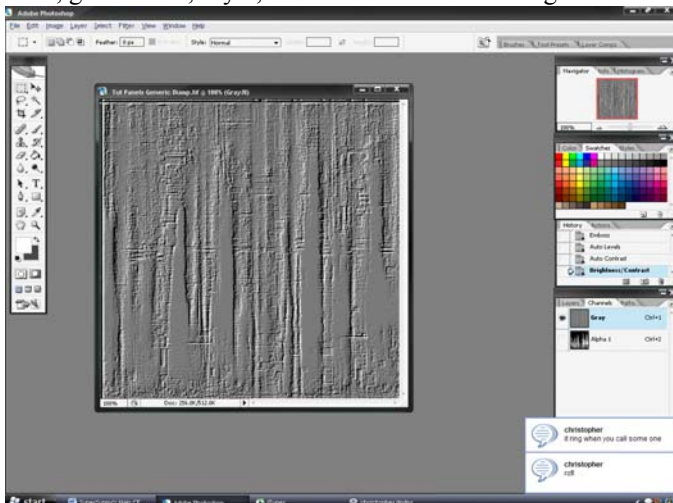
Your texture should be open.

Go to File, Save As, and save it as TextureNameBump.tif.

Exit your image and open up TextureNameBump.tif. Go to Image, Mode, and Grayscale. It is MUCH easier to work with bump-maps when they are black and white.



Next, go to Filter, Style, and Emboss. Make the angle about -32, and play around with the rest of the features.



radiosity properties

flags

☒ simple parameterization
☐ ignore normals
☐ transparent lit

detail level

high

power

0

color of emitted light

r 0

g 0

b 0



tint color

r 0

g 0

b 0



In the Flags section, if you're using a ground texture, I recommend flipping on "simple parameterization," helps a lot with lightmaps. Ignore the other two functions.

Detail Level: Keep it on low, but if you're going for a super "high" quality crazy styled map with unbelievably realistic lighting even though you'd have to wait a long time in radiosity, go to medium or high.

Power: The amount of power your shader gives off if you want it to give off specular lighting, like illuminate a certain area or something. I'd say around 15-25.

Color of Emitted Light: Exactly what it says. If you want me to explain this I should slap you.

Tint Color: Yeah I'm going to slap you REAL hard.

physics properties

material type

dirt

environment shader

Setting <true atmospheric fog> enables per-pixel atmospheric fog (for models) but disables point/spot lights, planar fog, and the ability to control the atmospheric fog density for this shader.

flags

☐ alpha-tested
☐ bump map is specular mask
☐ true atmospheric fog

Material Type: Go read a dictionary.

Alpha-Tested: If you're using advanced alpha maps. Don't worry about it for now.

Bump Map is Specular Mask: If you only want your bump-map to appear where specular light is. AKA-your flashlight does jack. (Not recommended.)

True Atmospheric Fog: It's for models. The model gets foggy with the fog...that didn't sound right.

environment shader type

Controls how diffuse maps are combined:

NORMAL:

Secondary detail map alpha controls blend between primary and secondary detail map. Specular mask is alpha of blended primary and secondary detail map alpha multiplied by alpha of micro detail map.

BLENDED:

Base map alpha controls blend between primary and secondary detail map. Specular mask is alpha of blended primary and secondary detail map alpha multiplied by alpha of micro detail map.

BLENDED BASE SPECULAR:

Same as BLENDED, except specular mask is alpha is base map times alpha of micro detail map.

type normal ▼

It literally explains it right there. Stop relying on me goddamnit.

lens flares

lens flare spacing 0 world units

lens flare ... Open

If you want your shader to have lens flares placed in it evenly by the length of how many world units you specify. This is how I make my glow effect happen with 0.12 spacing, and an edited lens flares. I'll go into that later.

diffuse properties

flags ☐ rescale detail maps
☐ rescale bump map

base map ... Open

detail map function double/biased multiply ▼

primary detail map scale 0

primary detail map ... Open

secondary detail map scale 0

secondary detail map ... Open

micro detail map function double/biased multiply ▼

micro detail map scale 0

micro detail map ... Open

material color r 0 g 0 b 0

Rescale Detail Maps and Bump Maps: If you're experiencing some difficulties and want to rescale them.

Base Map: All your base are belong to me.

Detail map Function: What kind of path your detail map will follow.

Primary Detail Map: Aww your mommy is here! AKA First Detail Map.

Secondary Detail Map: Am I your daddy now? AKA Second Detail Map.

Micro-Detail Map: Hey it's junior. AKA Mini Detail Map.

Material Color: What color you want your detail maps to change into.

bump properties

Perforated (alpha-tested) shaders use alpha in bump map.

bump map scale	0	
bump map		Open

texture scrolling animation

Scrolls all 2D maps simultaneously.

u-animation function	one	
u-animation period	0	seconds
u-animation scale	0	base map repeats
v-animation function	one	
v-animation period	0	seconds
v-animation scale	0	base map repeats

If you REALLY need to understand how this works, just IM me, because if you don't, that is SAD. It literally says what it is. Same with Self Illumination Properties right below it.

self-illumination properties

There are three self-illumination effects which are added together. Each effect has an <on color>, used when the shader is active, and an <off color>, used when the shader is not active. The self-illumination map is used as follows:

- * RED: primary mask
- * GREEN: secondary mask
- * BLUE: plasma mask
- * ALPHA: plasma animation reference

Each effect also has an animation <function>, <period> and <phase>, used when the shader is active. The primary and secondary effects simply modulate the <on color> by the animation value to produce an animation color, and then blend between the animation color and the <off color> based on the shader's activation level, and finally modulate by the mask.

The plasma shader compares the animation value with the alpha channel of the map (the plasma animation reference) and produces a high value when they are similar and a dark value when they are different. This value modulates the <plasma on color> to produce a plasma animation color, and the rest proceeds just like the primary and secondary effects.

specular properties

Controls dynamic specular highlights. The highlight is modulated by <brightness> as well as a blend between <perpendicular color> and <parallel color>.

Set <brightness> to zero to disable.

flags	☐ overbright ☐ extra-shiny ☐ lightmap is specular		
brightness	0	[0,1]	
perpendicular color	r 0	g 0	b 0
parallel color	r 0	g 0	b 0

reflection properties

Controls environment cube map reflections. The color of the cube map is "tinted" by a blend between <perpendicular color> and <parallel color> from the SPECULAR PROPERTIES above, and then modulated by a blend between <perpendicular brightness> and <parallel brightness>.

BUMPED CUBE MAP:

This type of reflection uses the shader's bump map (if it exists) to affect the reflection, as well as the perpendicular and parallel brightness (i.e. the "fresnel" effect).

FLAT CUBE MAP:

This is the fastest type of reflection. The bump map is used to attenuate the fresnel effect, but the reflection image itself is not bumped.

Clear <reflection cube map> or set both <perpendicular brightness> and <parallel brightness> to zero to disable.

flags	☐ dynamic mirror	
type	bumped cube-map	
lightmap brightness scale	0	[0,1]
perpendicular brightness	0	[0,1]
parallel brightness	0	[0,1]
reflection cube map	Open	

Overbright: OMFG MY EYES!

Extra-Shiny: Me and shiny, psshhh, we get it on alllll night.

Lightmap is specular: DO NOT click this.

Brightness: If you want your specular map to be really bright.

Perpendicular Color: Consult a dictionary for Perpendicular.

Parallel Color: Consult your mom for this.

Dynamic Mirror: Makes the shader reflective and dynamically mirrored. AKA Real Time Reflections. ONLY works on flat surfaces. If you get a "Could not finalize cluster mirror" error, that means you messed up.

Type: Don't make me come over there.

Lightmap Brightness Scale: If it's 1 that means you can see in the room even if it's supposed to be dark because of the reflection.

Perpendicular Brightness/Parallel Brightness: Jesus Christ!

Reflection Cube Map: Pick your cube map!

We are going to start by opening up levels\test\b30\shaders\Metal Panels Generic.shader_environment.

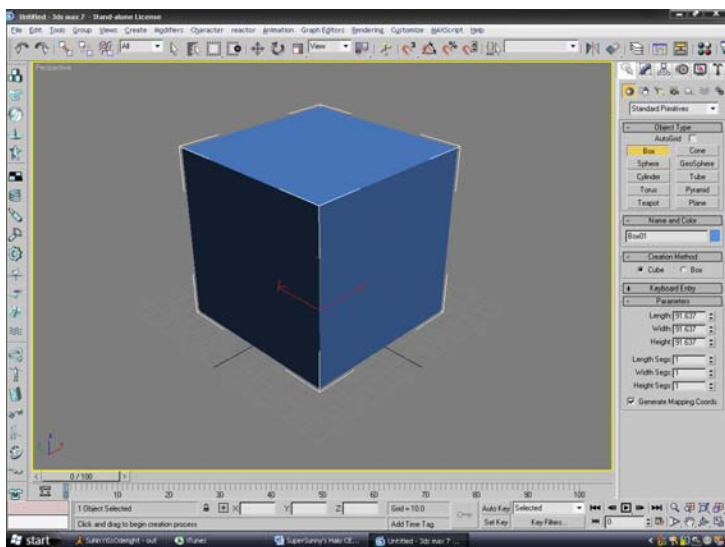
Now, go to File, Save As, and save it in your custom folder with a different name. Time to begin editing.

- Make the Detail Level low.
- Make the Base Map your custom bitmap.
- Make the Secondary Detail Map levels\b30\bitmaps\detail tech panels.bitmap
- Make the Bump-Map your custom bump-map.
- In Specular Properties, make it Over-Bright and Extra-Shiney. Change the two colors to correspond with a shade of your texture colors. I chose Purple and Blue.
- Make the Lightmap Brightness Scale 0.4.
- Make the cubemap whatever you want, but I chose levels\b30\bitmaps\cubemap metal tone.bitmap.
- Save it. Remember the name, you need it for the material in Max. Mine is called Tut Panels Generic.shader_environment.

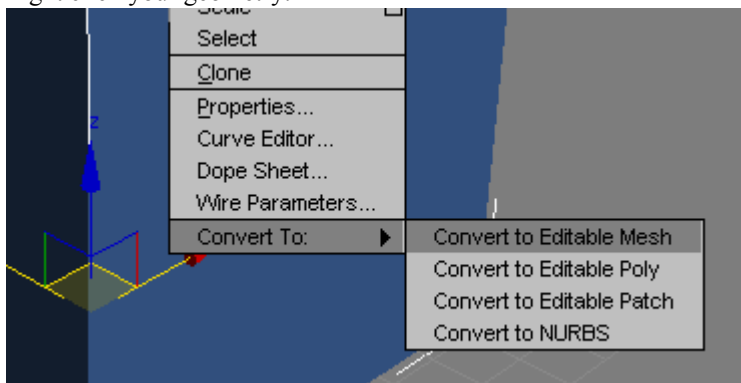
Chapter 2: Simple Box Modeling Setup.

Part 1: It's a Box!

Start up 3ds Max (or Gmax, but I'll be using 3ds Max for this tutorial).
Create a box. Whoohoo!



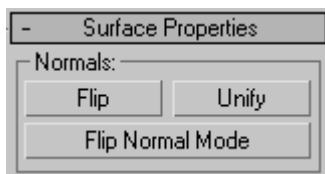
Now, flip all the Faces. (Here are the steps for those who do not understand.)
Right click your geometry.



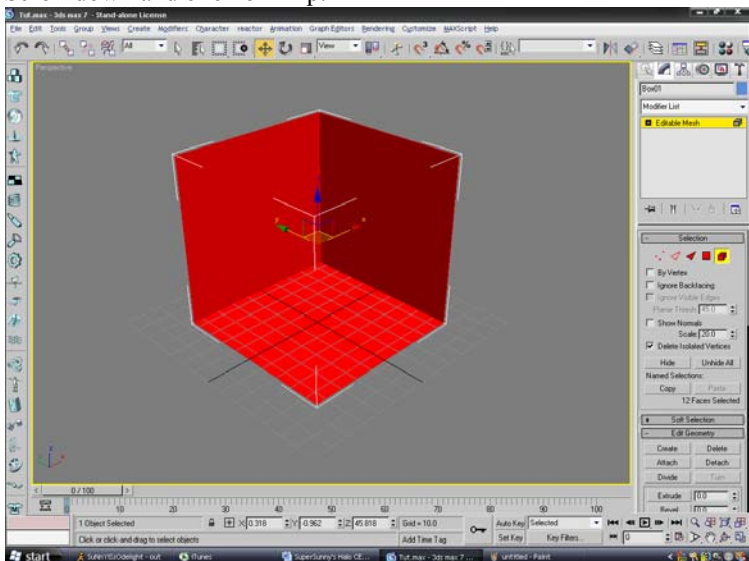
Convert to Editable Mesh.



Select the Element Tool on the Right.

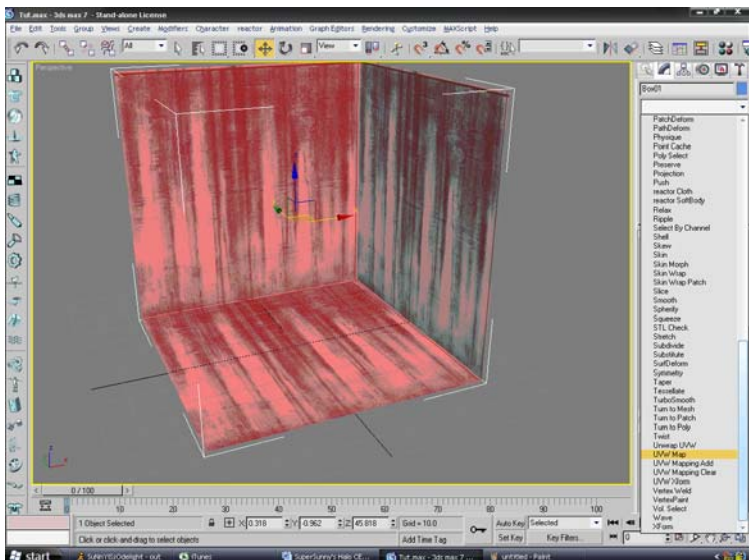


Scroll down and click on Flip.



.w0w: j1z fli994d! h4x?

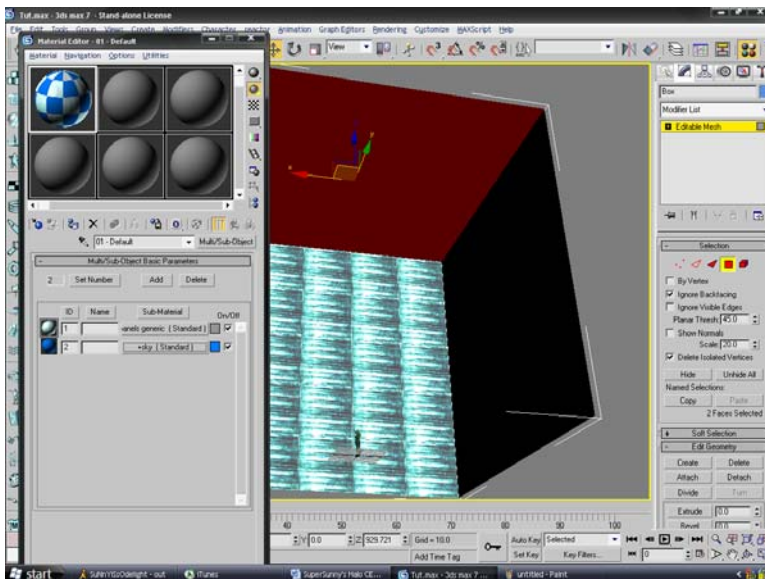
Did I really need to go through all those steps? Open the Material Editor (damnit read the HEK tutorial), and apply your custom texture with the proper material name to your model.



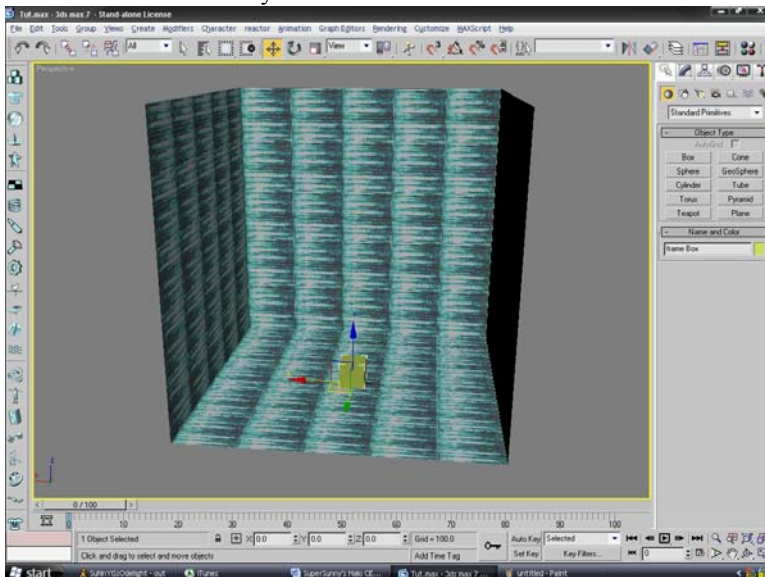
UV Map it. Select Box and give it about Tile it 5 U and 5 V.

Save. Import an MC Model if you wish to scale your model properly.

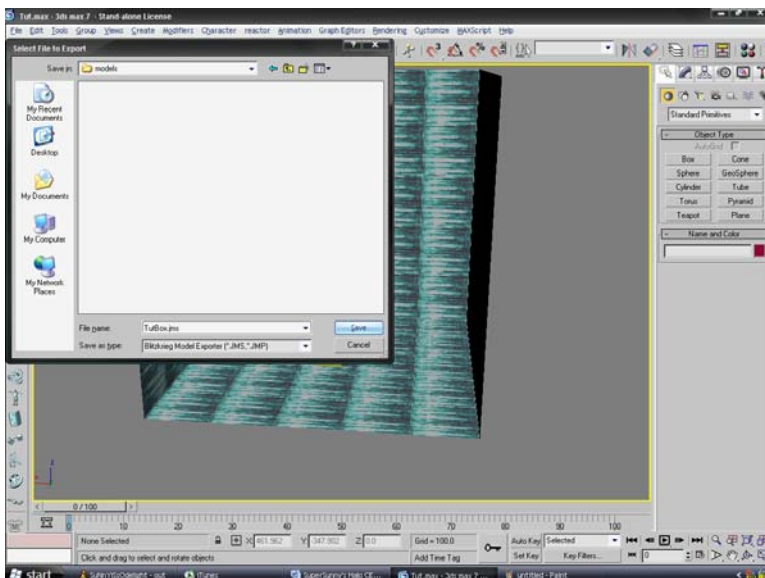
Now, select the top face of your geometry. Give it the material +sky so it becomes a skybox.



Create a frame and link your box to it.



Export.

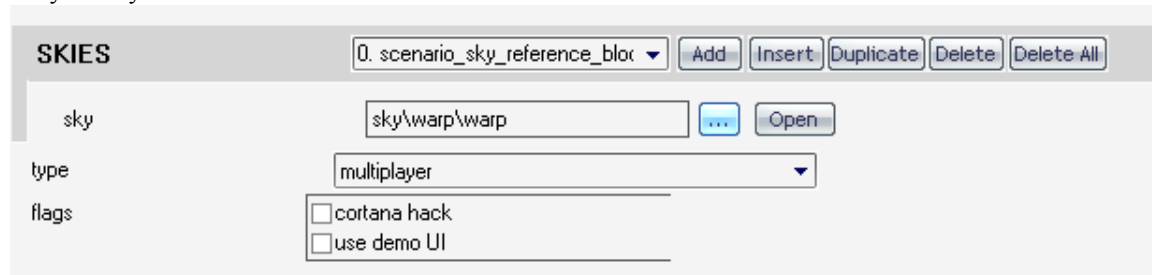


If you got any errors while compiling. I REALLY feel like slapping the hell out of you. Maybe even get Waffles to keep on IMing you, or make him call your cell while you're in Best Buy.

```
n>tool structure Tut TutBox
```

Structure it.

Set your skybox.



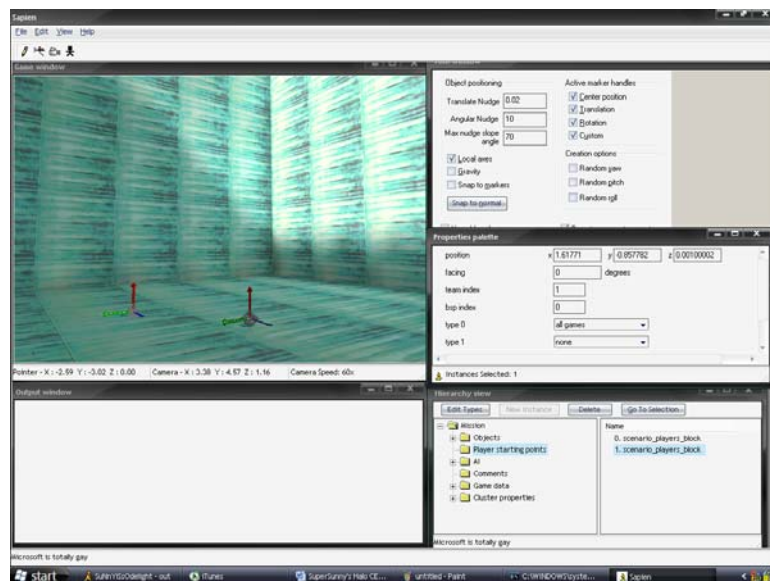
Radiosity it in Tool or Sapien, I'll be using Tool.

```
C:\Program Files\Microsoft Games\Halo Custom Edition>tool lightmaps tut\tut tutb
ox 1 0.000000000000001
Couldn't read map file './toolbeta.map'
i'm going to render final lightmaps for tutbox of tut\tut, and i'll stop when re
d+blue+green = 0.000000.
Hit control-c now if this isn't right.
```

Part 2: It's In!

It will go very fast, even on quality 1, because IT'S A GODDAMN BOX!

Open it in Sapien (if you already haven't.) Place the spawn points.



You can change the Perpendicular and Parallel Brightness in your shader if you want the cubemap to be shinier. Right now, it's got a cold-icy feeling to it.

Compile your map, and test it out in Halo. Use your flashlight and check out the bump-maps.

Now, you've gotten the beginning of your first sexy box in-game! Hooray! We haven't even started on the real lighting yet.



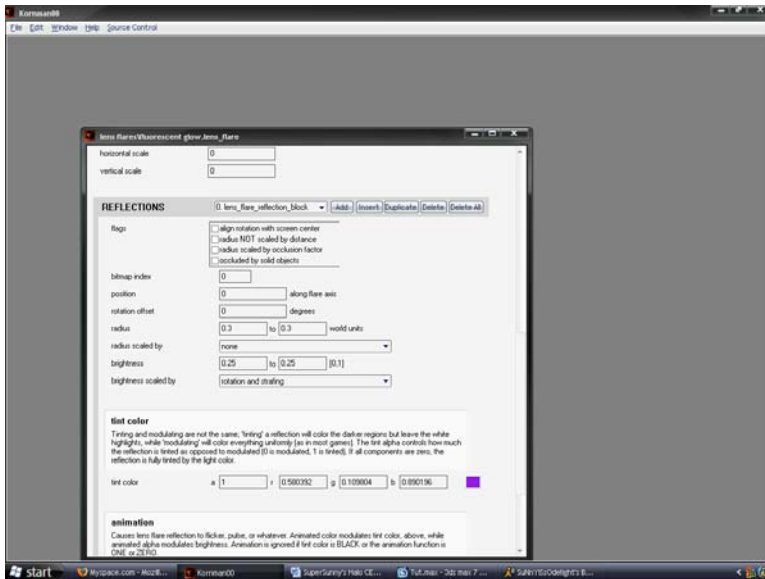
Chapter 3: Glow

Part 1: To Begin

Open up Max and Guerilla. In Max, open up your level. Select the faces of everything BUT the top and bottom. Tessellate Twice.

Minimize, and in Guerilla, open up Lens Flares\fluorescent glow.lens_flare
Go to File, Save As, and in your custom folder, save it as “Glow.lens_flare.”

In this tag, scroll down, and change the tint color. I used a mix of purple and blue.



There should be some checkboxes above it. Check the box that says “Occluded by Solid Objects.” Save.

Now, create a new shader_environment. Set the detail level to low. Scroll down to the Lens Flares section. Make the amount 0.05, and select your lens flares. Now, go back up, and set the Power to about 25, and select your lightmap colors. I’d choose the same ones I chose for the glow color. Set the material type if you wish. I chose plastic. Don’t worry about anything else, no need to be changed. Save as “GlowingShader.shader_environment” in your custom folder. In Max, select one face, anywhere. (a small one.)

Apply a material called “GlowingShader” to them, that’s all. Export, and structure it. You might have to delete the old structure_bsp from the bottom of the .scenario tag because sometimes Tool adds another one on instead of overwriting. Redo your lightmaps again.

Congradufucklations. Play around!