

Intermediate Blender: Rendering Off a Map

By Mr. D at Delta 3D

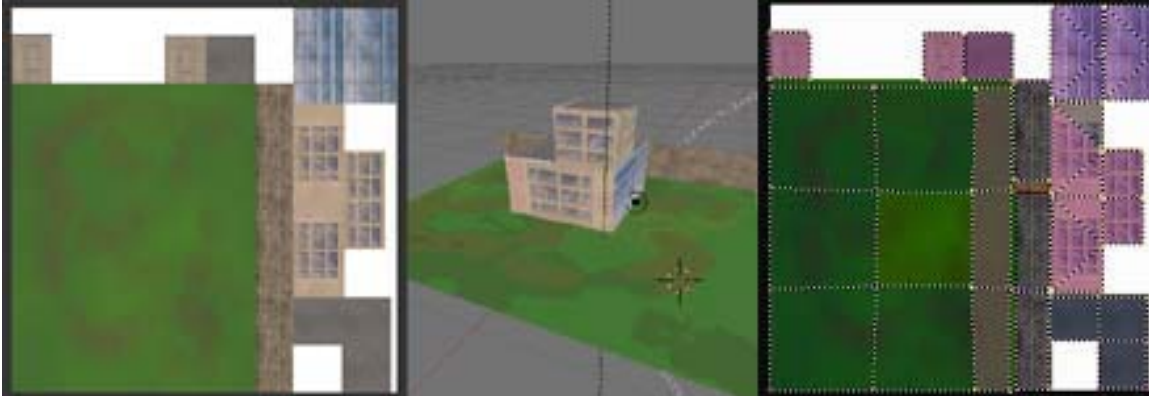
4-2007 Blender v2.43

NOTE!: This is not how to make UV texture baked radiosity maps, where Blender will take your mesh and recalculate mapping then lay it out for you. You must already have setup your mapping yourself for texture baking, the exception being how to lay out UV's for a simple lightmap as described below. This paper was written with OSG users in mind.

PART1) Rendering a texture

Mainly texture rendering is done to burn both lighting and other effects along with diffuse color into a single new map. This technique is useful when your object platform for a game project lacks the memory, and or power, to run real time lighting or even use lightmaps; or if your game engine is lacking in those lighting effects.

Artist often use this to help make textures as you can use the 3D program to create lighting and shadows in your textures for you. Plus as this is a render to the texture you can add a number of other channels such normal, specular, or shininess to your mesh and burn those qualities into your single texture.

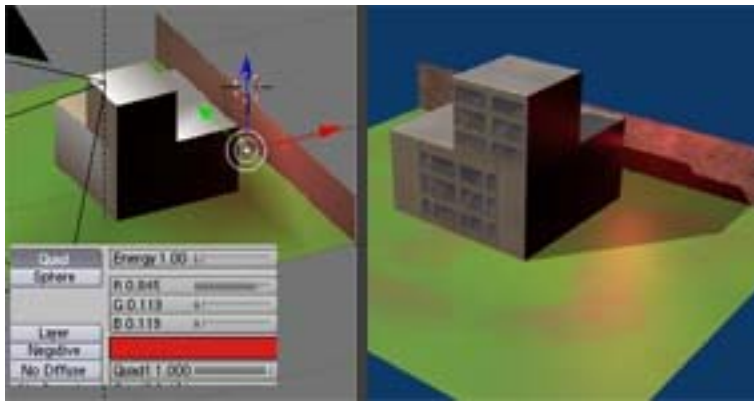


Any who read my OSG light mapping paper will recognize this model, and as I already laid out the UV mapping I thought I'd save myself some time. So look over to the left at the UV layout and notice how each face from the model gets its own spot on the texture. This can be one of the drawbacks to **burning in** textures, as to have good looking textures for an object with a large amount of faces you need an equally large texture. But for a character, especially one that usually will have all its faces laid out anyway, it makes burning simple.

If this were a level the artist would most likely go through and decide where they would benefit for burn in, and in others with flat lighting they might do no burning so they could say map a large area with a single repeating texture, or an area where UV mapping overlapped.

a) What do you want to use?

For this paper I'm just going to do a simple lighting setup, and add a bump map to Blender's Normal Channel to show how you can get effects burned in.



Ok I have one spot light far off to make my main shadow, and added a red lamp to light up one side of the building and brick wall.

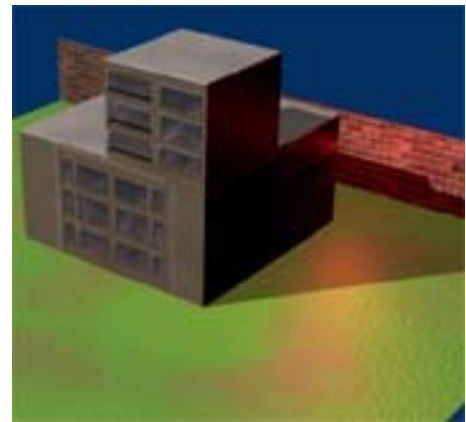
I find laying out lights easier in *Shaded* rather than *Textured*, but work as you like. Also render your image to

get a better feel for how the textures will look when rendered.



Here I grey scaled my original texture did some contrasting and added noise to the grassy area, and saved it off as a new file. Next I loaded it in to a *new Texture slot* of my Material and *mapped it* to the **Nor(mal) Channel** and increased the effects value slightly, resulting in my bumpy new look.

As the map is the same layout here as for my Diffuse Color, I could simply map my input for the normal map to the same UV's.



Rendering Out:

In a Buttons Window set it for 'Scene (f10)' and look for the tab 'Bake'. What you will want is to do a *'Full Render'*, as you are rendering lighting effects to the texture. It might be easy to think it would be the 'Textures' button, but that is for baking textures into an object for gaming.

As for 'Ambient Occlusion' try this link to understand it better.

http://en.wikipedia.org/wiki/Ambient_occlusion



And 'Normals' is for normal maps, and a quick explanation is faces have normals that are used to calculate shading in a 3d engine. In recent times it was found that a bitmap of one mesh's faces could be applied to another mesh, and the renderer would attempt to shade the object based on the normal layout of the first object. Main use being you take a normal map from a high resolution object and place it as a map on a similarly shaped lower resolution mesh. The engine tries to shade as it were the high resolution mesh, but the advantage is you do not need to constantly draw the high polygon mesh.

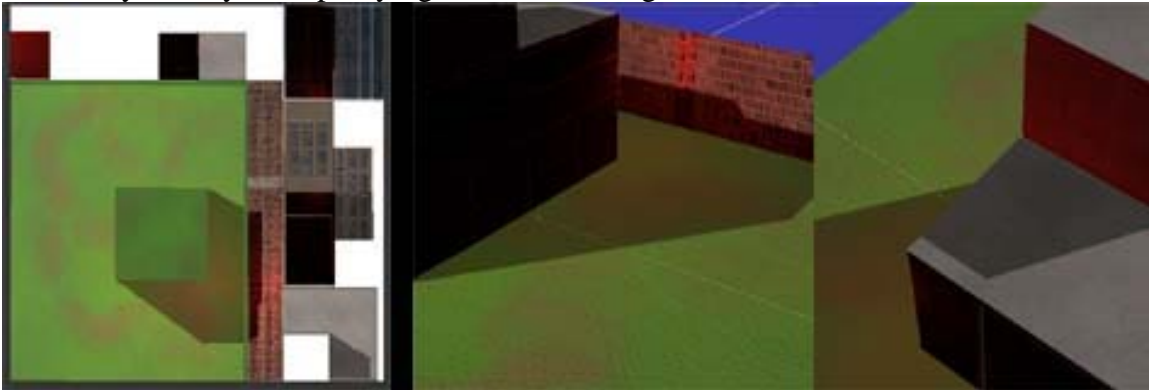
Clear: clears images from buffer before rendering.

Margin: adds an extended margin to UV map areas, if no space between UV's there is no margin.

Try different values to see what you like.

Once you've set what you want press the '**Bake**' button to carry out the action you've chosen.

Since my UV layout is pretty tight I'll use 0 Margin, Clear and Bake at '**Full Render**'.



Here is the result which shows in your **UV Editor Window**, next I did an *Image>Save As* from the UV window. **IMPORTANT!** Unless you are intending to save over your original texture, be sure to **use Save As** and *Not* just Save.

Finally I replaced the original texture with the new *Burnin* texture and exported out to the Delta 3D viewer now with lighting and bump effects in the texture.

Things To Remember

- 1) Each face gets its own UV space on the map, no overlapping UVs, the exception being if faces can share the exact same lightmap which rarely is the case. Example a long corridor where the ceiling always has the same light value, those faces could have just one tiny darkgrey lightmap for a dark ceiling.
- 2) If a complex mesh think about breaking it up so it might have more than one lightmap, as many most 3D programs will allow for multiple UV channels that can be mapped to groups of faces. Unfortunately the Blender OSG exporter is not currently able to handle this, though you might be able to edit an OSG file (see my Edit Blender OSG files for a starting point there).
- 3) The *Effects* are **burned in** so it's best to **turn off** OSG lighting for your mesh, also an OSG light will effect the mesh but just because your shining a light at a wall with a rendered shadow baked in will not make the shadow disappear.
- 4) If your target platform is slow and has little memory, **burnin** is one of the least resource intensive way to get some lighting effect.

5) If you're not the kind of artist whose good at drawing in shadows or specular highlights etc... Think about using this technique to get some lighting effect on your textures, start out by just trying slapping one bitmap on a quad, then make a simple grey scale picture to use in the Nor(mals) channel. Add a light and angle it, then render yourself a new texture.

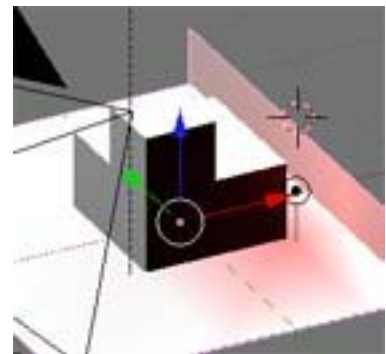
PART2 – LIGHTMAP

As you might have guessed making a light map is pretty much the same for a burnin map, only all you're interested in is the lighting intensity and color in your scene.



1- First thing you'll do is loose the textures from your material by in the **Buttons Window**> **Material**, 'Clear' out all textures in your material.

2- Locate your Col(or) box and change it to a near white. Why? Because remember though your loosing your diffuse map you can still use other effects like a specular map that could put highlights for lights on areas. If you color is full white there is no higher intensity



level you can go to for the specular highlights. Also recall this is just a bit map, and if it is not bright enough for your needs, you can always brighten it in a paint program. To the right is my model without its textures, just the lighting.



MAKING A NEW IMAGE

Next you want to make a new map to save out your image on. So *why not* do this for map burnin' instead of worrying about do a *Save As* like above. Blender needs the Bitmap for diffuse texture to show in the UV Editor to understand what the original map to burn into is. It is not looking at your Material Texture list.

But for your lightmap no texture is needed just a color.

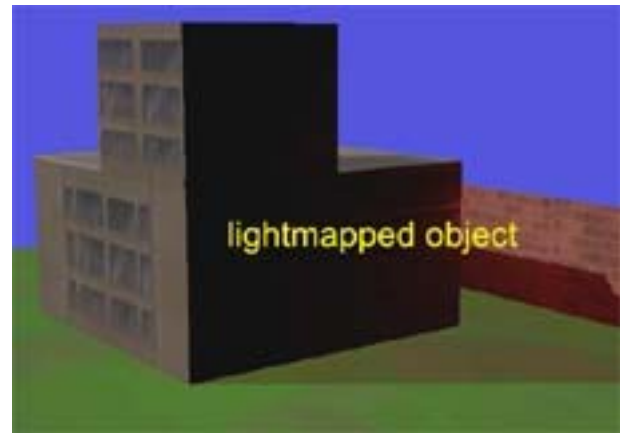
In the UV Editor look at the Menubar for the Image hotspot and do **Image**> **New**. A pop up will appear in which you can name and size your new image, take into account if for gaming this is a power of 2 for size. And hit OK. It will look like just a blank black area now in the UV Editor but that's alright, as that is your new texture as the Blender programmers just decided to start with a black image rather than a white one.



Now go back to the **Buttons Window**> **Scene**> **Bake Tab** and do your '**Full Render**', and save off your new lightmap.

For those reading this working with Blender OSG you have to hand edit in your lightmap to your .osg exported file of the diffuse object; as before see my paper on editing.

For people working with the Blender engine in your material make a new texture slot, load your new lightmap into it and map it to your '**emit**' channel.



HELP MY IMAGE HAS OVERLAPPING UV'S AND TILING

Using Blender 2.43 to make a second UV channel, and to layout a mesh so each face has its own UV spot.

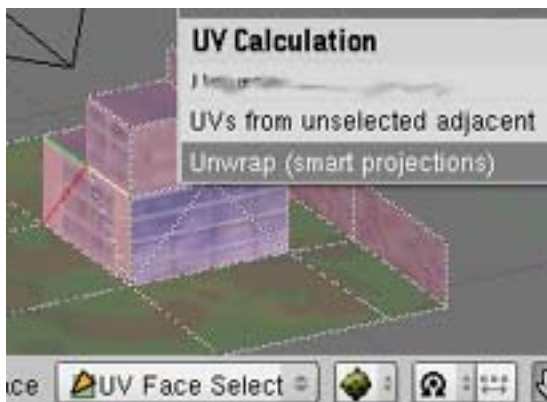


In the **Buttons**> **Editing** Window, look for the **Mesh Area** and locate '**UV Texture - New**' button and press.

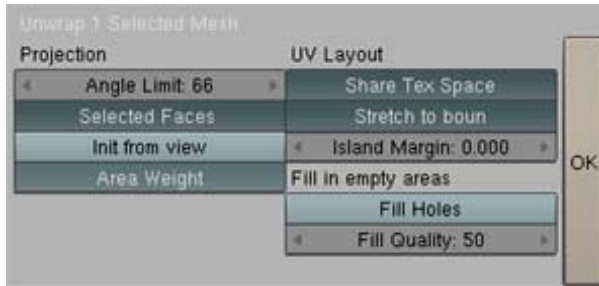
You now have a second UV Channel to work with, and to tell which is active look the left side and see which one has the **dark colored square**.

To get rid of a UV channel simply press the **X** at the right side.

For now pick the new channel.

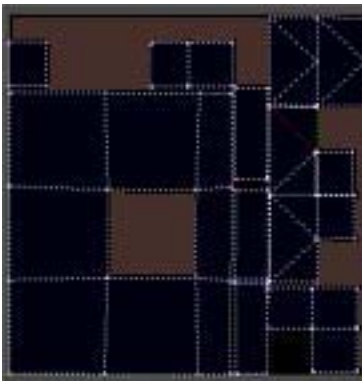


Next you'll recalculate UV's for your mesh, so select it and go to **UV Face Select Mode**. Select all faces and hit '**U**' to bring up the UV Calculation pop up, and at the bottom of the list choose **Unwrap (smart projections)**.



Smart Projections pop up.

For now just hit OK, but later on try changing the '*Angle Limit*' and such. Also take note of the 'Selected Faces' button, meaning different groups of faces can be selected for such things as the use of multiple maps.



Here is the results, where each face gets its area, the dark reddish were colored by me to indicate unused areas of the map. And you will notice some faces get a small area, now some of those areas may contain places where you might want more space allocated for a shadow/lighting effect.

This UV layout can be edited as usual so you can move or scale this mapping to your liking. You are just using *Smart Projections* as a quick way to get faces their own space, it does not have to stay this way.

IMPORTANT!

As above this will not redo your texture and make one that matches the new UV layout. It only creates a new set of UV's, Blender is working on this type of UV texture baking and hopefully soon it will be available. There is a plugin that some people are having some success with that maybe out there now.

So for OSG users go to your first UV channel with your Diffuse setup, make sure diffuse bitmap is reloaded, and export an .osg file.

Next switch to your second UV channel and follow the steps for making your lightmap, but you will need to save this file out as an .osg as well. Why? Because while you may use a cut and paste of the *Texture Unit* from the Diffuse model output, changing the *unit number to 1* and changing the bitmap listed to your rendered and saved lightmap image.

What you need this file for is its *TexCoordArray* numbers found in the .osg file, but kindly see the editing file to understand Array number matching problems in the Editing Paper.