Lightmapping in Maya
for Unity

Lightmapping in Maya

Lightmapping in Maya

In this tutorial we'll be going through the steps of taking a simple scene in Maya and Lightmapping it for use in Unity. Lightmapping is also known as Prelighting or Shadow Mapping. It is a great way to add realism to a scene while also cutting down on some of the processing overhead in your game by reducing the number of runtime lights you will need to use in your Unity Scene.

We're going to take a modeled, textured scene and perform two passes onto a secondary UV set:

- An Ambient Occlusion pass.
- A Ray Trace pass.

This tutorial assumes only a general understanding of Maya's layout and workflow so even beginning users should be able to get through it easily enough while referring to the Maya manual. As with most complex applications, there are usually a dozen different methods that can be used to achieve the same results. This tutorial covers what I consider to be the fastest and easiest way to achieve Lightmapping for use with Maya/Unity.

First, let's start a new Project in Unity. Include the Standard Assets Package and import the LightMappingTutorial.unitypackage into the new Project. This will bring in a folder called TutorialAssets. Open the TutorialStart Scene inside this folder.

At this point we have a pretty drab looking house that would only fit into a game called "Another Rainy Day in Seattle". So let's get to spicing this scene up a bit. Double click the House

Prefab to open it up in Maya.

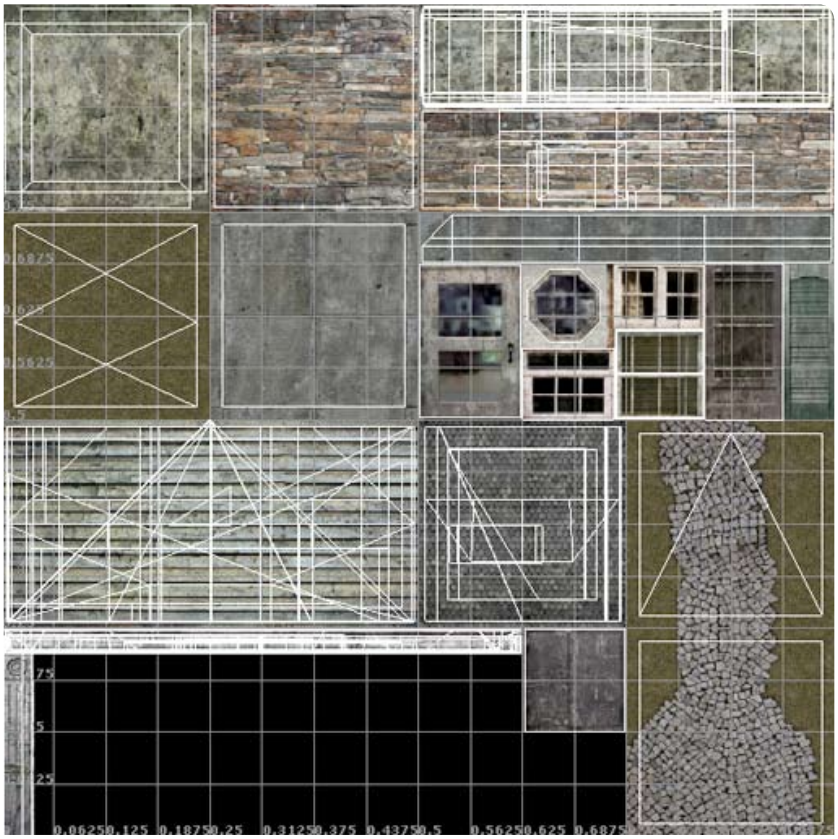
I have included an FBX version of this tutorial for anyone that wants to take a look at the finished tutorial, but who does not use Maya. It can be found in the FBXVersion Folder inside TutorialAssets.

Once the House Prefab file is loaded into Maya, the first thing we notice is that the whole scene is only made up of three meshes: “HouseMesh”, “TreeTrunks”, and “TreeBranches”.

The scene is split up in this way because each mesh will perform a different function not only for Prelighting the scene, but also in determining how the Shaders and Mesh Colliders will work in Unity.

Each mesh is on its own Layer in Maya so we can easily hide them for certain parts of the Prelighting process. We also only have three Materials, one for each type of mesh. This is fairly self explanatory for the trees, but for the house it’s worth explaining the importance of combining all the textures needed for the house into one large texture sheet: In Unity, and with most game engines, it’s of the utmost importance to combine as many meshes as possible together and to have each mesh only have one Shader assigned to it to get the greatest performance. The more you can combine meshes and textures the better the performance will be.

Select the HouseMesh and open up the UV Texture Editor (Window/UV Texture Editor...) to take a look at how the model is UV mapped.



As you can see the house model is UV mapped in a way to tile certain sections of the texture. A lot of UVs overlap to use the same part of the texture sheet over and over again. Because so many UVs overlap on this model it's necessary to create a second UV set for the lighting.

Let's start adding the Lightmaps to this scene, starting with the Ambient Occlusion pass.

Ambient Occlusion is a method for faking the look of Global Illumination. It creates soft self-shadowing in all the little nooks and corners of an object that helps to more clearly define the contours while also bringing a certain softness and overall coherency to it. Ambient Occlusion is calculated by casting rays in every direction from the surface of the object. Wherever a ray hits another surface the origin point of that ray is darkened on the surface. All these points are then averaged out to create soft, dark areas where light would be naturally occluded.

To create our first Lightmap we need to make a new Material:

- Open up the Hypershade Window (Window/Rendering Editors/Hypershade...)

- Create a new Surface Shader (Create/Materials/Surface Shader)
- Create a Mental Ray Ambient Occlusion Texture (Create/mental ray Textures/Mib_amb_occlusion).
- Double-click on the "mib_amb_occlusion1" node to bring up the Attribute Editor and rename it "AOTexture".
- Rename the "surfaceShader1" to "AOShader".
- Middle-button-drag the AOTexture onto the AOShader and choose Default as the input connection in the contextual menu.

  If you only have a two-button mouse, right-click on the output connection of the AOTexture node and select outValue and right-click on the AOShader and choose Default as the input connection. Then go buy a three-button mouse immediately because Maya is painful to use otherwise! From here on out I'm going to assume you are using a three-button mouse.

We now have our Material to make the Ambient Occlusion pass for the HouseMesh.

- Double-click the AOTexture node in the HyperShade Work Area to bring up the Attribute Editor again. We only need to change 2 settings for the AOTexture:
  - The first setting is the number of Samples to use. The default is 16, but in order to get some nice soft shadowing we need to bump this number up a bit. At 16 the Ambient Occlusion will be quite splotchy. Increasing the number of samples can exponentially increase the rendering time so it's best to bump this number up just enough to get a good result. For this scene 64 samples is a good compromise; if you're more patient, or have a fast computer then by all means bump it up a bit more to 128!
  - The second setting to change is Max_distance. If set to zero (the default), the occlusion

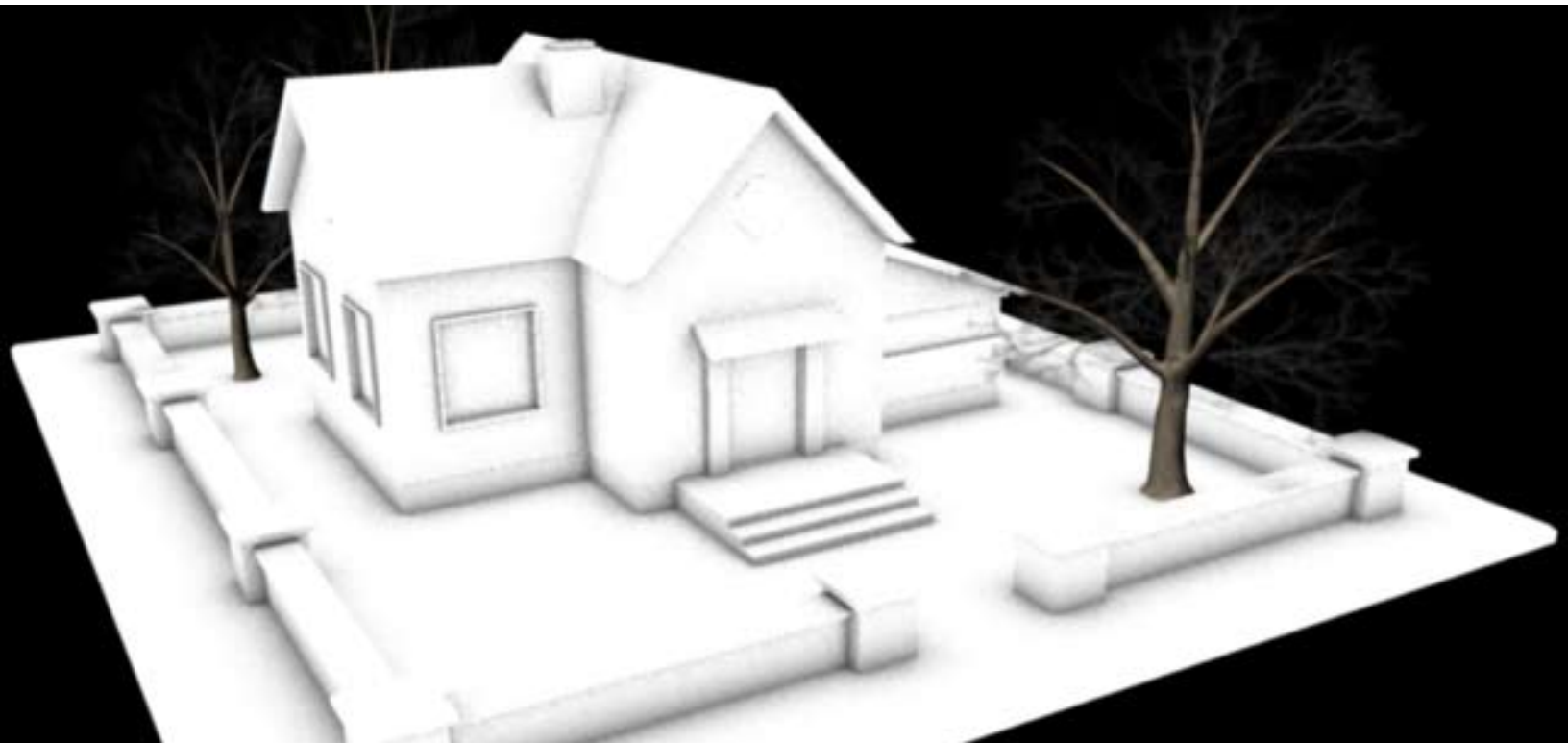
rays will shoot off to infinity. This is unwanted because we don't want surfaces far away from each other to have any effect. For instance, the tops of the trees shouldn't darken the grass. As this scene is pretty much to scale, let's set Max Distance to 1.5. That's all we need to set up for the Ambient Occlusion Shader.

- Finally, middle-button-drag the AOShader onto the HouseMesh (it should turn black).

Let's make sure everything is working so far by making a test render to see the AOShader in action.

- First we have to switch the renderer to Mental Ray so go to Window/Rendering Editors/Render Settings...
- At the top, change Render Using to "mental ray".
- Do a test render.

If everything is set up correctly, you should get a render similar to that in Fig. 3, below. There is some slight splotchiness to the Ambient Occlusion (depending on the amount of Samples you used), but it will not be noticeable when combined with the other textures.



Now that we know it's working, let's get to the important part of baking this off to our first Lightmap.

We need to create a new UV Set for our Prelighting passes. The simplest way to do this is to use Automatic UV Mapping.

- Hit F3 to make sure you're in the Modeling menu.
- Select the HouseMesh and go to Create UVs/Automatic Mapping (options).
- Set the number of Planes to 6
- Set Optimize to "Fewer Pieces"
- Set the Scale Mode to "Stretch to Square"
- Set "Shell Stacking" to "Shape". This ensures that Maya will map the UV shells into the tightest composition.
- Finally, check Create New UV Set and name it "lightmap".
- Hit Project.
- Now open up the UV Texture Editor and take a look at the 2 UV sets we have by going to Image/UV Sets.

We now have a UV set named "map1" and one named "lightmap".

We're now going to "bake" this Ambient Occlusion pass:

- First, hide the "TreeBranchesLayer" (click the "V" on that Layer to hide it in the "Channel Box/Layer Editor"). We need to hide this layer because the branches have an alpha channel

that isn't taken into consideration when calculating Ambient Occlusion. If you left the Tree-Branches visible it will calculate only the polygons and create unrealistic shading.

- To bake off this Ambient Occlusion pass select the HouseMesh and go to Color/Batch Bake (mental ray) (options). There's a ton of stuff to set up in here:
  - Objects to Bake is "Selected"
  - Bake to Texture—make sure Bake Shadows is set.
  - Camera is "perspective".
  - Color mode is "Light and Color".
  - Check "Orthogonal Reflection (without this checked, the AO calculation will pinch every corner creating unwanted shading).
  - Normal Direction is "Surface Front".
  - Prefix is the name of the texture that we're creating -- name it "HouseAO".
  - Set the Resolution to 2048 x 2048 and the File Format to "TIFF"
  - Check Bake To One Map
  - "Fill Texture Seams" is really important: It expands the texturing beyond the UV shells by a certain number of pixels. If set too low you will get ugly black seams in your bake, but if set too high the texture will start to bleed between UV shells. For a 2048 texture like this "10" is a pretty safe bet.
  - The last setting is to check Override mesh UV set assignments and to name it "lightmap".
- Now hit Convert and Close and go get a snack, or read "War and Peace"... your mileage may vary, but either way this is going to take some time!

And we're back... but oh no, what happened here! This looks horrible—it didn't bake to the right UV set!

Have no fear, we're just at the ugly duckling stage. The HouseAO is being drawn onto the "map1" UV set, but it actually calculated the Ambient Occlusion using the "lightmap" UV set. For now all we're concerned with is that the Ambient Occlusion got baked out to a texture file.

- Drag the HouseDiffuse shader back onto the HouseMesh.

It's time to set up our lights for the Ray Trace pass:

- Create an Ambient Light by going to Create/Lights/Ambient Light (options).
- In the Options window, set the Intensity to 0.15 and the Ambient Shade to zero. This light is used to adjust the contrast of our shadows, if you want lighter shadows then bump up the Intensity a bit.
- Uncheck Cast Shadows and hit Create.
- Next, create a Directional Light.
- Open up the Attribute Editor. Change the Color to a pale yellow (try HSV (48.35, 0.185, 1.0)).
- Go down into the Shadows tab and check Use Ray Trace Shadows.
- Now rotate the Directional Light to a more interesting angle. For example, I've rotating mine to (-45, 45, 0)—an afternoon sun.

The last thing we have to do is set up Mental Ray:

- Open up the Render Settings Window (Window/Rendering Editors/Render Settings...).
- Click on the "mental ray" tab and change Quality Presets to "Production".
- Now go down to Raytracing. Make sure Ray Tracing is checked.

- Go down to Shadows and change the Shadow Method to “Sorted”. This is important because sorting takes into account the Alpha Channel on the TreeBranches. Without this Mental Ray will render out the shadows of the TreeBranches as just polygons.

In Maya 2008 a bug has popped up where artifacts will appear when using Raytracing with textures that use an alpha channel. To fix this we need change one property in the “Branches-Material”. Open up the Attribute Editor for this Material, expand the “Raytrace Options” pane and set the “Shadow Attenuation” to Zero. If you are using an earlier version of Maya changing the Shadow Method to “Sorted” (named “Sort” in earlier versions) should be sufficient.

Ok, we’re now ready to bake off the Ray Traced Shadow Pass...

- Make sure the “TreeBranchesLayer” is now visible and select the HouseMesh.
- Go back to Color/Batch Bake (mental ray) (options). Time to change these settings again:
  - Change the Color Mode to “Only Light”.
  - Rename the Prefix to “HouseShadow”.

Everything else can stay the same.

- Hit Convert and Close and this time just get up and have a quick stretch because this render isn’t going to take nearly as long as the last one.
- When the bake is done open up the HyperShade window and rename the Material that was just created by the baking process to “FinalShader”.

Ok, now it's time to combine our 2 passes into 1 texture...

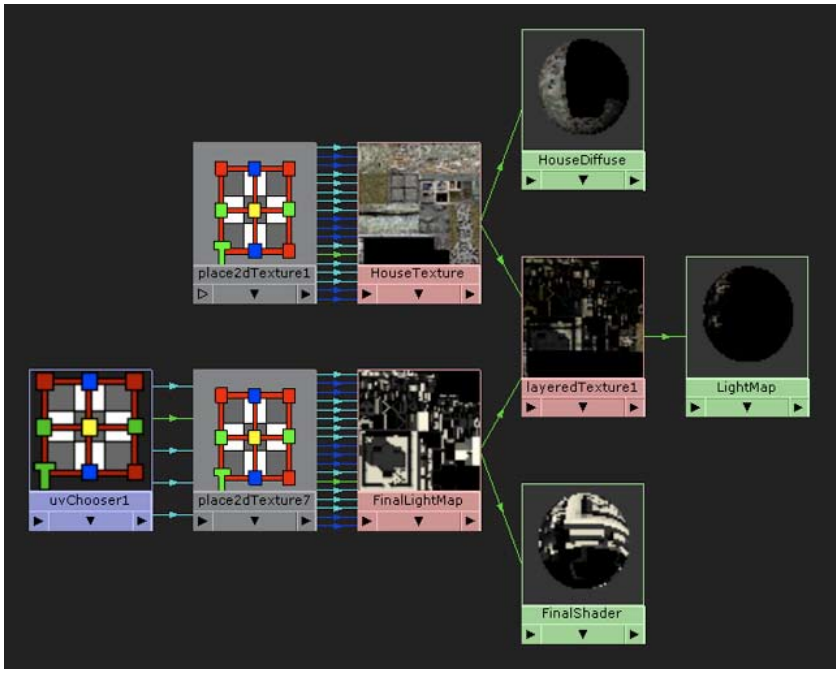
- Open up PhotoShop (or whatever image manipulation app you're using) and open our 2 texture files. They can usually be found in "Documents/maya/projects/default/renderData/mentalray/lightMap". (If they're not at that location then you've set up a different default Project location for use with Maya.)
- Copy/Paste the "HouseAO.tif" image into "HouseShadow.tif".
- Make sure HouseAO is the top layer and change its blending mode to Multiply. This will nicely blend the 2 light maps together. I usually duplicate the Ambient Occlusion layer to increase the contrast a bit.
- Save out the new file to "LightMapTexture.psd" inside our Unity Project's Asset Folder, making sure to preserve the layers because then you can go back and tweak how each layer affects the shading.
- Go back to Maya's Hypershade Window and create a new Lambert Material and name it "LightMap".
- Clear the Work Area by going to Graph/Clear Graph.
- Make sure the "LightMap" Material is still selected and go to Graph/Input Connections.
- Create a Layered Texture and connect it to the default input of the "LightMap" Material.
- Right-click on the "FinalShader" and in the contextual menu, choose Select Input Nodes.
- Now go to Graph/Add Selected to Graph.
- Repeat the process for the "HouseDiffuse" Material.
- Go to Graph/Rearrange Graph to clean things up a bit.

Now we need to load the combined texture we created in PhotoShop...

- Select the “FinalShader” and bring up the Attribute Editor.
- Select the File tab, change the name at the top to “FinalLightMap”, click on the folder icon and load in “LightMapTexture.psd” from your Unity Project Folder.
- In the Work Area, middle-button-drag the FinalShader’s texture node onto the Layered Texture node and choose “inputs[n].color/[0]”.
- Middle-button-drag the HouseDiffuse texture node onto the Layered Texture and choose “inputs[n].color/[1]”.
- Bring up the Attribute Editor for the Layered Texture. Click on the leftmost blue box and change its Blend Mode to Illuminate. (Multiply also works and in some cases looks better).
- Now drag the “LightMap” Material onto the HouseMesh.

Ouch, it still looks like a mess! That’s because the UV sets aren’t properly assigned...

- Right-click on the HouseMesh and choose UV Sets/UV Linking... in the contextual menu. A Relationship Editor will come up with UV Sets on the left and Textures on the right.
- Click on map1 and make sure it’s assigned to the “HouseTexture” by clicking on it in the right-hand column.



Now make sure lightmap is assigned to “FinalLightMap”.

Presto! Everything should look alright now. Save the file and switch over to Unity.

In Unity...

- ...select the “HouseMesh” and switch the Shader type to “Lightmapped/Diffuse”.
- Drag the “LightMapTexture.psd” asset onto the “Lightmap (RGB)” slot.

Looking good so far! Finally, we need to create a Directional Light in Unity that replicates the Directional Light we used in Maya...

- Create a new Light Game Object/Create Other/Directional Light.
- Rotate it to the same angle as the Directional Light we used in Maya. (The only reason we need this is because the trees need to match up to the HouseMesh lighting.)

Alright, that pretty much wraps it up! I hope this tutorial has taught you enough to create your own Light Maps in Maya. If you have any technical questions about this topic I’m more than happy to answer them, just PM me on the Unity forums at aNTeNNa trEE.