EA official tutorials:
Terrain editor
Leveleditor
Custom Mod Setup
Understanding Heightmaps
Build the Bunker - 3ds Max
Build the Bunker - Maya
Import the Bunker
Build the Car - Maya
Import the Car
Create new team - Maya
Create new team - 3ds Max
Create new kits - Maya and Max

Mapping - General
Setup FH2/bf2editor
FH2 mapping guide
FH2 mapping standards

Mapping - Texturing
Texture system in bf2
Terragen skies
Terragen texturing

Mapping - Heightmaps
World Machine
Geocontrol and bf2hmt

Mapping - Lightmapping
Simple lightmap setup

Mapping - Downloads
Sky settings bf2
Water settings bf2
Lowdetailtextures/settings bf2

3D modelling/texturing
Skinning the Daimler Dingo

Coding

FORGOTTEN HOPE 2
Mapping tutorials

Battlefield 2 Modding Tutorial 4 - The Bunker - Maya version

by [EA]Lawrence Brown

Download Maya files here

Now that we've created our basic mod, it's time to actually start making the assets to put in it.

We're going to start out with a simple building: the bunker. This will be an all-purpose building, the base providing cover fro enemy fire with window slots allowing soldiers to shoot back, and also a small tower providing a better view of the battlefiel a vantage point for laying cover fire and sniping, and also a potential platform for mounting static weapons such as machine guns and TOW rockets.

Set Up:

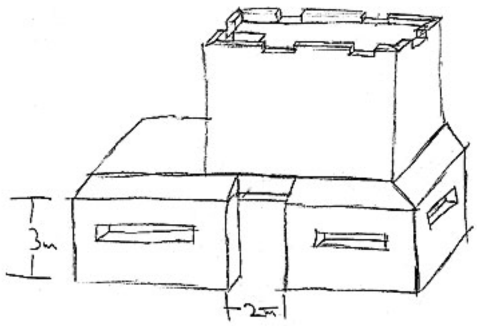
Before we begin, do the following:

1. Open the "Tutorial Files" folder that came with this document and follow the ReadMe for proper installation of the tutor files. They are required in order for this tutorial to function properly.

This tutorial also requires that you have installed all the Maya tools that shipped with the editor.

Modeling:

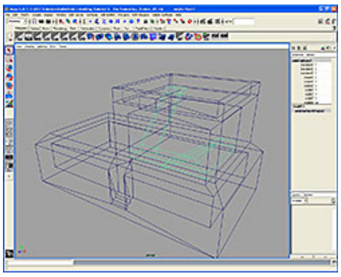
We're going to start with a simple sketch of the building. This could come from a designer on the team, or it may be someth you draw up yourself. The point is, even if you are just modeling for yourself, it's good to at least draw out a basic form on paper to help visualize what you're trying to do. Here's our sketch for the bunker:



Now that we've got a blueprint of sorts, we can begin modeling.

The first step is to create the main visible LOD (Level of Detail)

1. Launch maya and open up My_Bunker_01.mb. Here I've roughed out the shape of the bunker, creating the interior and exterior and a system of ramps to get to the top. Notice that I haven't put in details like the windows or the crenalatio at the top of the tower (the notches in the sides of the walls). At this point we just want the basic shape so that we car export and test the overall design. Since we will probably have to make changes, there is no sense wasting time on details that would have to be reworked later.



2. Select the two pieces and combine them into one mesh by selecting “Polygons>Combine” from the menu bar.
3. Rename this piece “My_Bunker”.

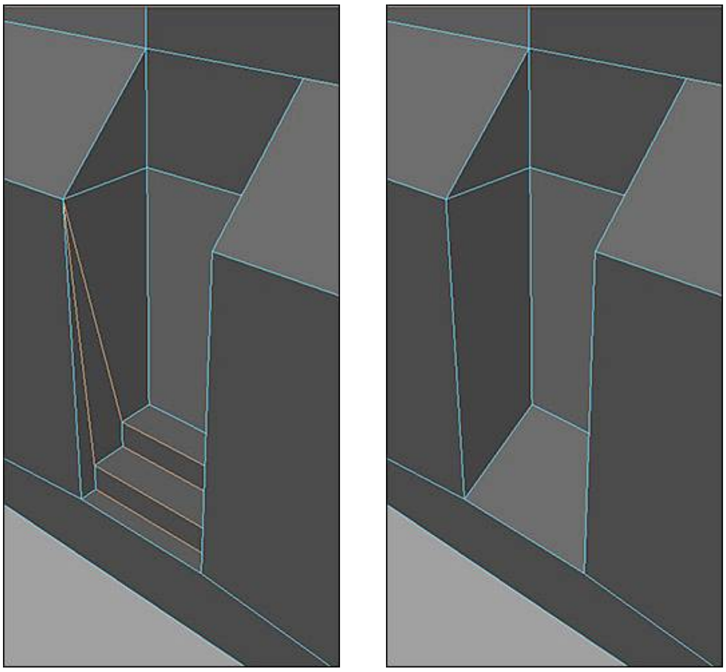
Col0:

The next step is to create the collision meshes. First is the projectile collision mesh, Col0.

4. Duplicate My_Bunker and rename it “Col0”.

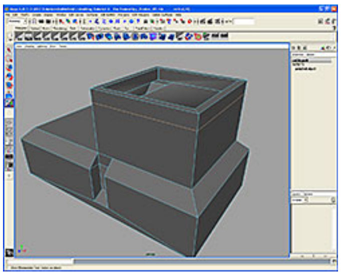
Since this is the projectile collision mesh, we want to optimize it by deleting any small details that we don’t need. If there w things like antennas or other small items attached to the building, we would delete these. We also want to simplify the mesh where possible. The less polygons the game has to test against collisions, the faster the server will run. This building is currently so simple that there isn’t much to optimize. Even so, we should still do the following.

5. Hide My_Bunker so that we can clearly see Col0 and zoom in on the stairs. Remove all excess edges until the stairs become a ramp like the following:

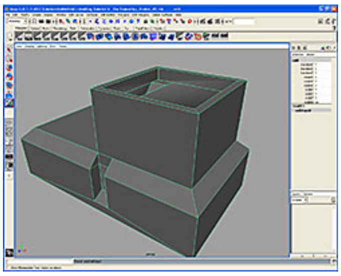


Note: make sure you remove any extra vertices that may have been left behind.

6. Now select the edges highlighted in orange in the following screenshot and remove. The edges go all the way around tl tower, so make sure you select all four:



When you’re done, the Bunker should look like this:



Col1:

We've completed the optimization of Col0. The next step is to create the soldier collision mesh, Col1.

7. Duplicate Col0. Rename the new object "Col1" if it doesn't do so automatically. (Note: "Col1", not "Col01"!)

Normally we would do further optimization on this mesh, but there is really nothing to remove since it's so simple. More abo this later.

Col2:

The last collision mesh to make is the vehicle collision mesh, Col2.

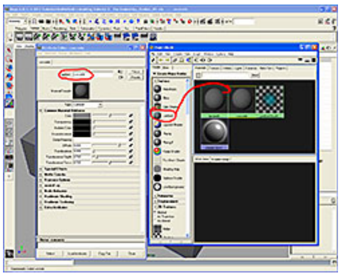
8. Following the same steps as earlier, duplicate Col1 and rename it "Col2" if it doesn't do so automatically.

This mesh would also be optimized, but we're not going to waste time on it now since it's just for preliminary testing. Keep i mind that we will be going back and adding more detail to the visible LOD once we are satisfied that the overall design works At that time we will also be deleting the current collision meshes and creating them again from the updated LOD . Reworking objects is just part of the modding process, so it's best to get used to it now.

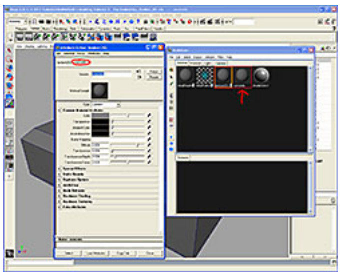
Texturing:

The next step is to texture the pieces we created. You may either continue your work or open "My_Bunker_02.mb" for this section.

9. Open the Hypershade and with the middle mouse button, drag a new Lambert shader to the materials window.
10. double-click on the new Lambert shader to open up the attribute editor and change the name to "concrete":



Note: If you are using the Multilister instead of the Hypershade, make sure you are changing the name of the Lambert shade to "concrete", not the Shader G roup! If you use the Multilister, it should look like the following when you're done:



You can change the color of the shader to anything you like. The color is just to make it easy for you to identify. All the importer is concerned about it the name.

11. Assign the concrete shader we just made to the three collision meshes.

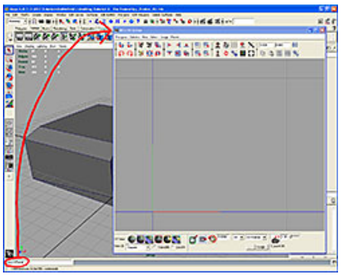
Concrete is the name of one of the materials in the game, so what will happen is that the importer will read this name and apply the material to the collision meshes. This is why it's so important that it is spelled right. For a complete list of materia consult the Material Editor section of the BF2 Editor.

The Visible Mesh:

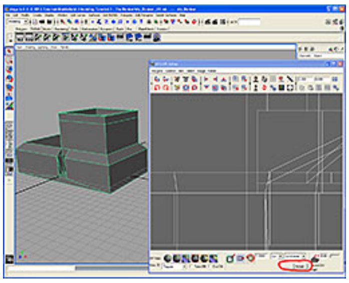
Now we have to texture the visible mesh. Since this is just a test and we know we are going to add more detail later, we are just going to apply a simple placeholder shader for now.

12. Hide everything but My_Bunker so that it is easier to work on.

13. Enter “dnyUVPanel” in the command line and press enter (or click on the shelf button if you’ve made one) to open up t BF2 UV Editor:

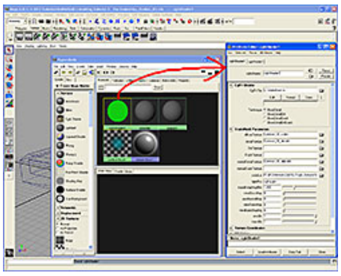


14. Make sure that My_Bunker is selected, then in the UV window press the “Assign” button:



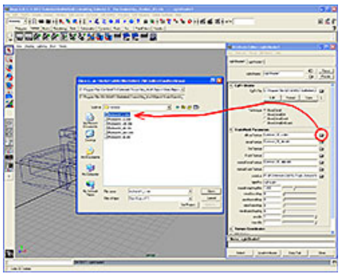
What this does is creates a cgFX shader and assigns it to the selected object. It also should create a pointlight that is connec to the shader that you can move around to see how the lighting affects your normal maps. (Sometimes the light isn’t createc it isn’t, don’t worry about it. It won’t affect anything.)

15. Open up the Hypergraph window and double-click on the cgFX shader you just created to open up the attribute editor:



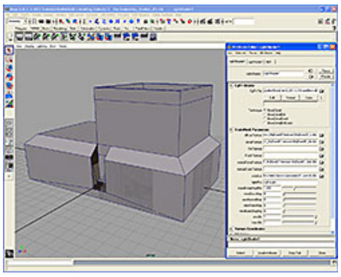
We need to assign some basic textures so that the bunker exports properly. There are a lot of options in this shader, but we' get more into that in a later tutorial. Right now we just want the basics.

16. In the “StaticMesh Parameters” section, click on the folder icon to the right of “diffuseTexture”. In the window that pop up, select “C:\Program Files\EA G ames\Battlefield 2\bf2editor\RawData\Objects\StaticObjects_MyBase01\textures\MyBase01_c.dds”:

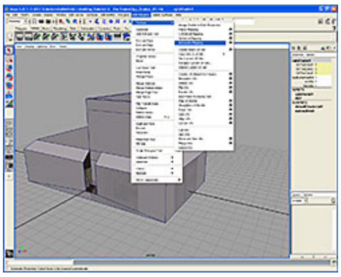


17. Do the same for “detailTexture”, choosing “MyBase01_de.dds” from the same folder.
18. Do this a third time for “normalDetailTexture”, choosing” “MyBase01_deb.dds”. (“_de” stands for detail and “_deb” star for detail bumpmap.)
19. Finally, click on the folder icon next to “colorLut” and select “C:\Program Files\EA G ames\Battlefield 2\maya\C G FX6.0\cgfx_textures\SpecularLUT_pow36.dds”.

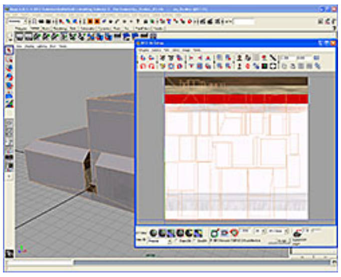
If you press “6” on your number pad to go into Textured-Shaded mode, your bunker should look something like the following screenshot. Don’t worry if the texture placement is different because we haven’t set up the UV’s yet:



20. The editor and the BF2 engine don't like UV's with zero surface area, so make sure My_Bunker is selected, then choos "Edit Polygons>Texture>Automatic Mapping" from the menu bar. We don't really care what the options are, but you m want to reset them to default if you've used this tool much. This is just a quick way to make sure that the UV's of eac polygon have some surface area:



Your UV's should now look similar to this. Don't worry if your texture looks strange or is all green. We'll fix that in a minute. Just worry about the UV's for now:



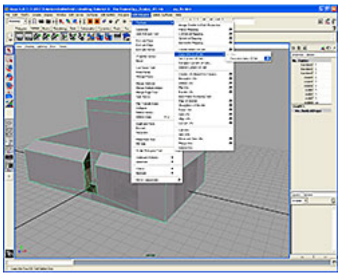
23. Unhide the collision meshes and do the same thing to each of them. Even though the collision meshes aren't seen in the game, the engine doesn't deal well with polygons with zero surface area when calculating collisions either. Automatic mappir will take care of this. It doesn't matter what the actual coordinates of the UVs are, only that they have some surface area.

Creating the Second UV Set:

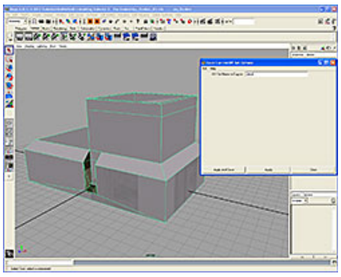
Now that we've laid out basic UV's, the last thing we have to do texture-wise is create a second set for the visible mesh. Mo on this in a later tutorial. For now you just have to know that the diffuse texture in the shader needs one UV set and the deta texture needs a second one.

It's important to note that only the visible mesh needs the second UV set. The collision meshes should only have one. If the collision meshes have more than one UV set, it could crash the game!

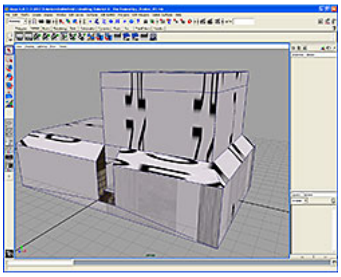
21. Hide the collision meshes again and reselect My_Bunker. Select "Edit Polygons>Texture>Copy UVs to UV Set>Copy int New UV Set>Options" from the menu bar:



22. In the window that pops up, enter "detail" in the text box and click "Apply and Close":



It should now look something like this:



Again, if your textures look different, it doesn't matter. When you do the automatic mapping, Maya may sort the polygons differently depending on the options chosen.

Triangulation and Cleanup:

We have finished the texturing, but there's one more step we have to do before we can put everything together. The importe doesn't read anything but 3-sided polygons on the collision meshes, so we have to fix this now.

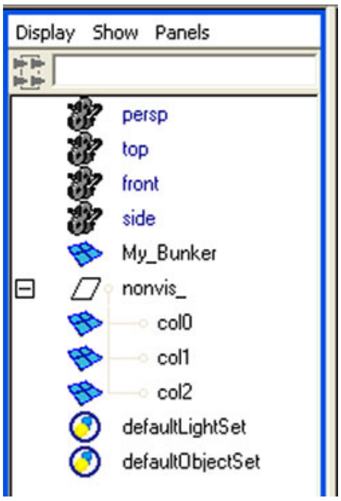
23. Unhide the collision meshes if you can't see them and select all three. (just the collision meshes, not the visible mesh)
24. From the menu, select "Polygons>Triangulate". Maya will triangulate the mesh and switch to face selection mode.
25. Click somewhere off of any of the objects to deselect everything, then make sure that everything is visible and drag-select to select everything in the scene.
26. Delete history and freeze transforms if necessary to be sure we have clean geometry. You are now ready to put everything together.

Setting up the hierarchy:

The last step is to set up the hierarchy for the importer to read. It's very important to follow these steps exactly because the importer reads certain information from the various names and expects things to be in a certain order. Make sure there are spelling mistakes and to pay attention to where there are capitalized letters.

You may either continue with the scene you've been working on or open "My_Bunker_03.mb.

27. Unhide all parts. You also may wish to open the outliner to make it easier to organize things.
28. Select the three collision meshes and group them.
29. Rename the new node "nonvis_". Note the single underscore at the end of the name:



(Sorry the hierarchy lines are so hard to see. You may have to save the picture and open it in photoshop and adjust the leve you are having trouble seeing them.)

30. Create a locator and rename it "My_Bunker__Anchor".

Notice that there is a double-underscore in the middle of the name. The text before the double-underscore is just the name o the object and can be anything you want. Anything after the double-Underscore is read by the importer and is used to create the appropriate code in the editor. In this case, “Anchor” is basically a pivot point, but can be moved around to tweak an obj For now just leave it at the origin.

31. Select the “My_Bunker_Anchor” locator and the “nonvis_” node and parent them both under “My_Bunker”.
32. Select “My_Bunker” and group it to itself. Rename the new node “lod0”.

In a later tutorial we will be adding more LODs. For now we only need this one.

33. Now with “lod0” selected, group it to itself again and name it “geom0_LightmapSize512”.

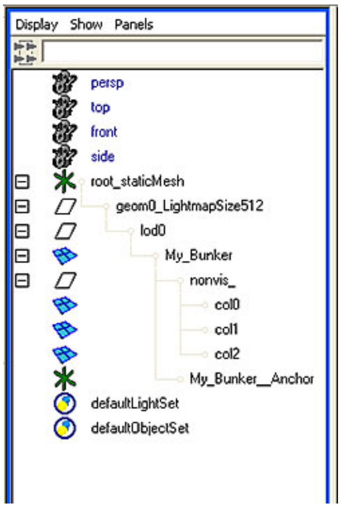
This is more info for the importer to use. In this case it tells the importer to create a lightmap texture sheet that is 512x512 pixels.

34. Create a second locator and rename it “root_staticMesh”.

All static objects have one of these. There should only be one in each Maya scene because this is what the importer looks fo first. If there is more than one, it will get confused.

35. Parent “geom0_LightmapSize512” underneath “root_staticMesh”.

Your hierarchy should now look like this:



```
persp
top
front
side
root_staticMesh
    geom0_LightmapSize512
        lod0
            My_Bunker
                nonvis_
                    col0
                    col1
                    col2
                My_Bunker__Anchor
defaultLightSet
defaultObjectSet
```

36. Save your file as “My_Bunker.mb”. (Not My_Bunker_04.mb! This is our “final” version, the one that we will import into the editor. You don’t want any numbers at the end of the file, or the object will end up being called My_Bunker04 inste of My_Bunker. Numbers at the end of a name may also confuse the game and cause strange results.)

You have now completed your first staticMesh. In a later tutorial we will import this into a custom level.

If you have any problems or want to compare your version to the finished version, open up “My_Bunker_Complete.mb”. You can use this file as reference for setting up basic staticMesh objects.

You are now ready to move on to “Battlefield 2 Modding Tutorial 5 – Importing the Bunker”.