Lef's Intro To Modding (Blender)

Foreword: This isn't intended to be a comprehensive guide and just a starting point to get into modding with blender. I'm still learning many things myself.

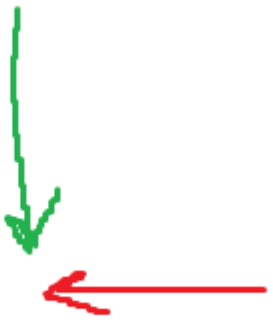
Textures

So with an item in FF you have 2 things; the model and it's textures. The model is just your mesh with all its polys, edges, and faces and whatnot. The textures are 2D images that map to the 3D model using something called a UV map (I'll get onto this later).

You will come across different textures in FF and they behave slightly differently depending on what they are used for.

Normal Maps:

These are the purply looking textures. Normal maps show how light should interact with your model and can give extra detail without the extra polys. This is useful because it takes far less work to render a normal map and "fake" the detail rather than rendering it through the mesh. It will basically give the effect of bits sticking out or inwards.

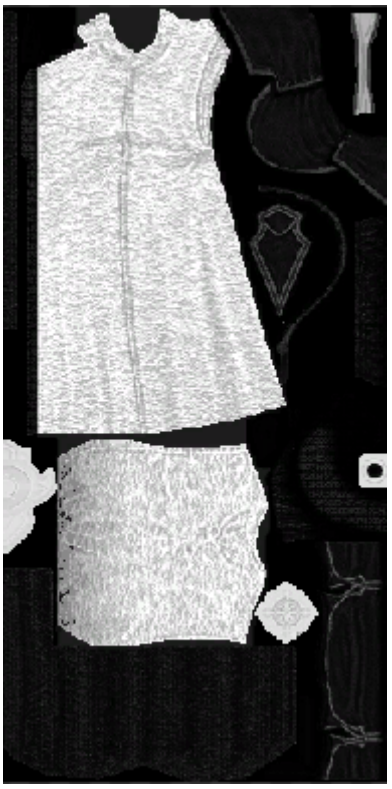


Normal maps use RGB colours. The red and green channel give the information for the light direction, with the blue channel function varying depending on the texture use. The red channel is the light coming from right to left. The green channel is the light going from top to bottom. The blue channel on the standard shader corresponds to transparency in the texture. The alpha channel on the normal map corresponds to the colourset values.

Specular/Multi Maps:

This is another one that has varying effects depending on the shader used.

When used alongside a diffuse on a standard shader they will typically look black and white with high contrast. Whiter areas will render with more “shine” while darker areas will be less shiny.

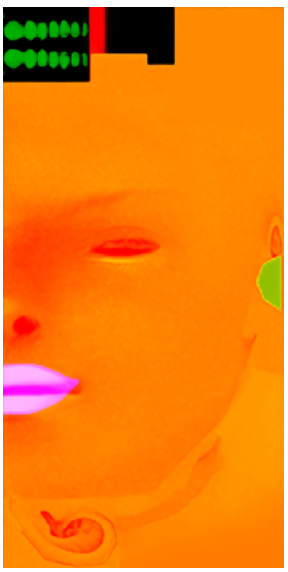


When used as a multi with a colourset on a standard shader they will typically look white/green. Greener areas will have shadows, with darker green equalling darker shadows. Whiter areas will have less/no shadows.

When used with a hair shader on the etc textures, multis will look like this:

The information in the colour channels in this case tells the game how to colour and shade the various components on the texture. For example, the pinkish area is the red channel and corresponds to the tattoo on female midlanders. Any area coloured with this will also dye with tattoo colour. Eyebrows have a strong green component with a barely noticeable red component. Eyelashes are basically the same as eyebrows however they lack the red component. You can play around with this multi and get various effects such as dyeing lashes or eyebrows with tattoo colour or hair colour.

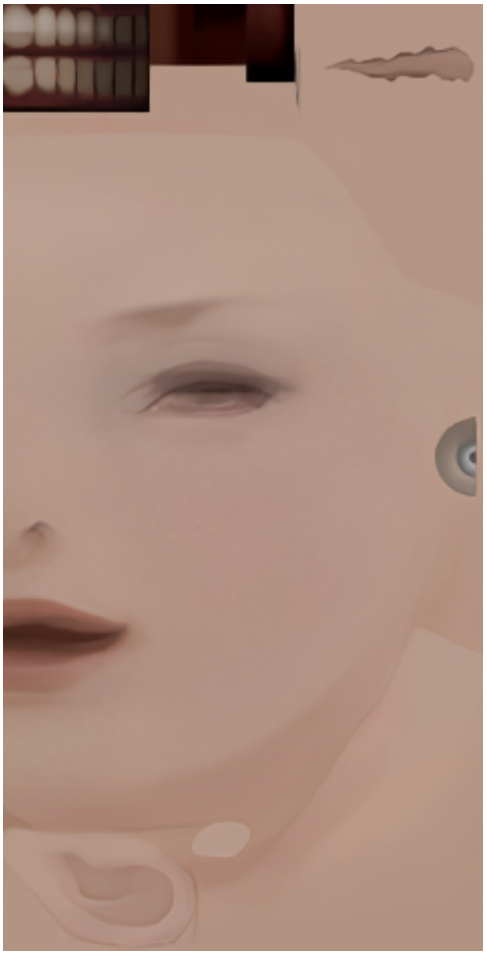
When used with a skin shader on the face textures the multi looks like this:



The blue channel corresponds to lipstick colour and the red channel corresponds to the skin colour.

Diffuse Maps:

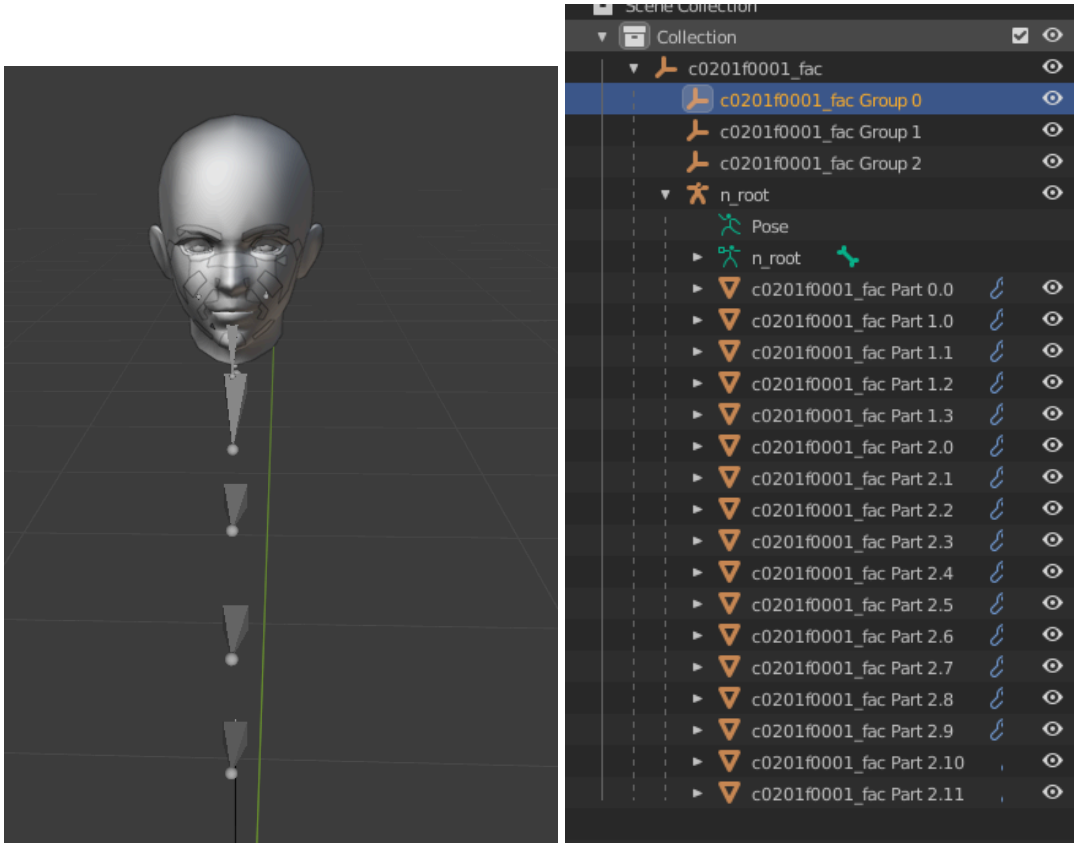
Just straight up a picture projected onto a mesh. This is typically the texture that makeup is done on.



Models

Start by importing your FBX file into blender.

Here we can see the various mesh parts and the armature.

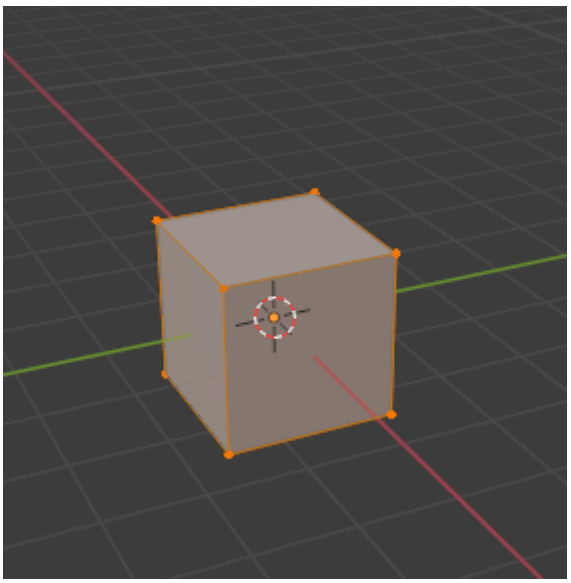


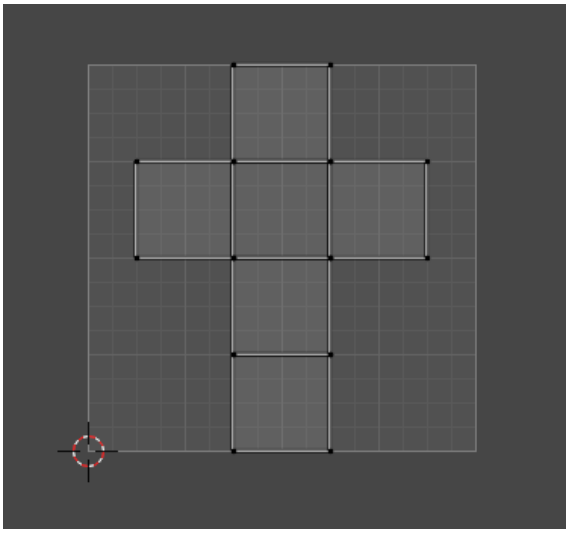
There are some things you need to be aware of and keep in mind to be able to create a successful mod. These are; the mesh, the armature, parenting, weights, and the UV map.

UV Map

To connect the earlier segment to this segment, I will explain the UV map. Put simply, it's like a 2D cutout of your mesh.

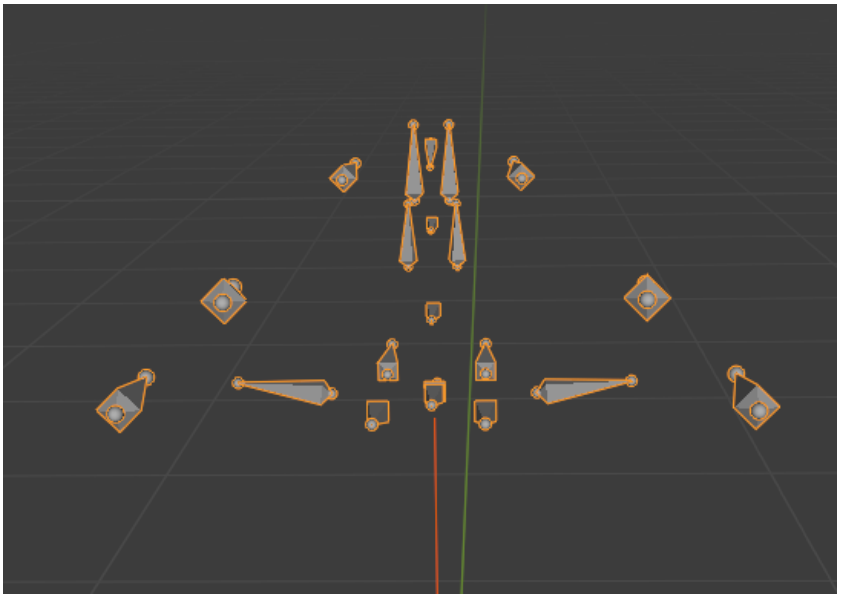
It is difficult to project a 2D plane onto a 3D object. This is where the UV map comes in. Your 3D object will need to be unwrapped (cut out any lay flat) and the UV aligned appropriately in order for the texture to display correctly on it. This is easy to visualize with a cube. Here we can see how the faces of the 3D cube map onto a 2D plane. FF vanilla meshes will come pre-unwrapped.





This document explains where to see your UV maps:
https://docs.google.com/document/d/1hdjFzNqeKjqj0ulx_d75MndcCVXggAnqlXCTCGHhz5Y/edit

Armature, Parenting, and Weights

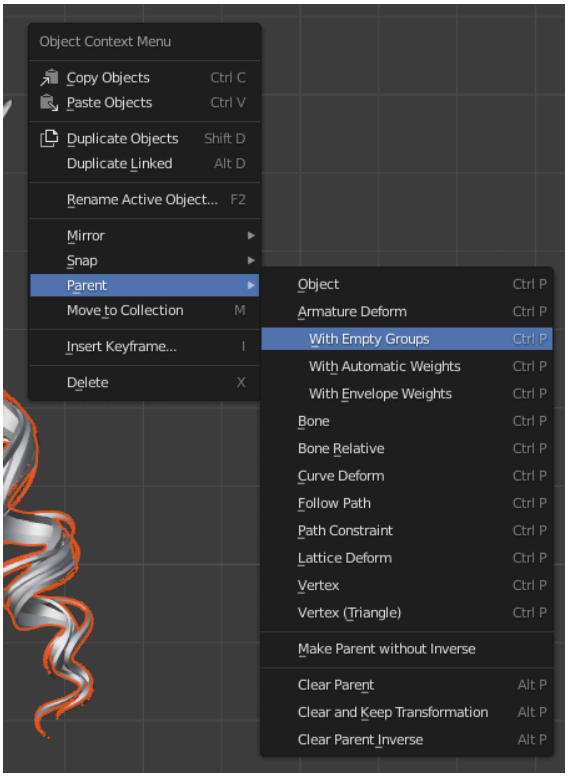


Also called a rig or skeleton, these are the first layer in what animates your objects. If a mesh is rigged correctly, the animation will dictate the movement of the bones and hence move the mesh in the desired way. The individual pieces that you can move are called bones.

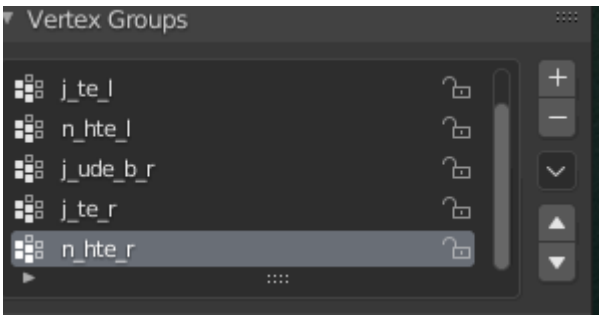
In order for the bones to exert an influence over the mesh, the mesh (the child) must be parented to the armature (the parent) and have weights painted on. Parenting means that

the child object will follow the parent object depending on what the objects are. Here it just means that the mesh will follow the bone movement according to the weight influence.

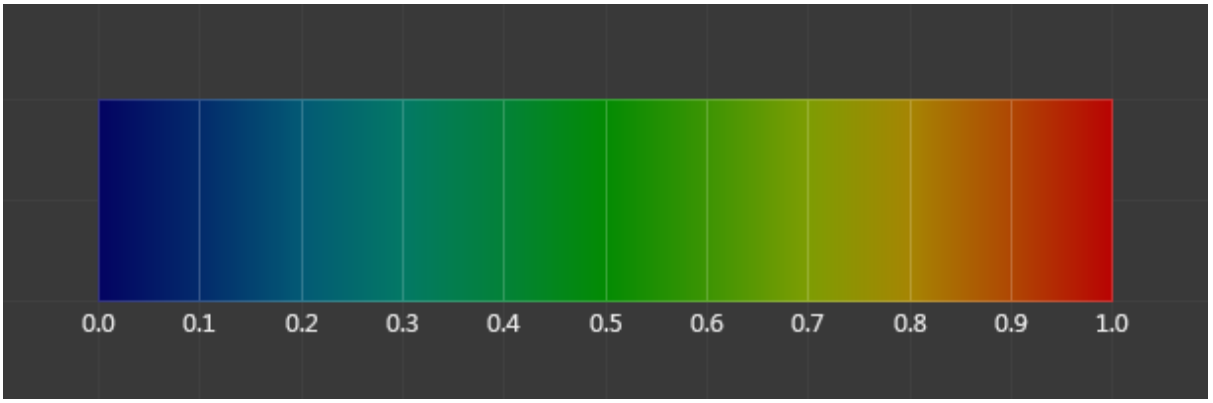
To parent your mesh to the armature; select your object/mesh, then shift click your armature. Right click and select parent with empty groups. This will parent your mesh to the armature but also add vertex groups for each bone. If there are any weights attached to these vertex groups already they will be preserved. This is useful for when you are wanting to move several modular pieces to one armature.



In the context of modding in this instance, each bone that has weights on the mesh will create a vertex group.



Weights are a numerical value given to a vertex, dictating how much influence that vertex group/bone will have over it. In blender this is represented by a heat map, with dark blue having no influence(0%), and red having full influence (100%).



FF has a limit of 4 bones per vertex, so when you are working and ready to export you will want to keep this limit and the weights normalized for all vertices (they all add to 1.0).

To get into weight paint mode; select your armature then shift click your mesh/object. Select weight paint from this dropdown: