

# Assignment 3: Textures

CS148 Fall 2015-2016

## Introduction



Figure 1: A 3D model with diffuse and specular textures.

This assignment will build upon the previous assignment to also incorporate textures. In the provided assignment framework, we show you how to use a diffuse and specular texture to perform Blinn-Phong shading. You will expand upon this to implement normal mapping which allows relatively low-poly meshes to look like they have a lot of detail on them. Textures are commonly used to make low-poly models look much more realistic, especially in video games where it is cheaper (to a point) to render a high quality texture on a low-poly mesh than a normal-quality texture on a high-poly mesh. Remember, start early and ask questions if you get stuck!

Throughout the assignment, you should also go back to the scene you setup in assignment 2 and start texturing it to submit as your final image.

## Useful Resources

1. OpenGL Programming Guide Chapter 6 (the book is not required, but recommended!)
2. OpenGL 4 Documentation: [here](#). Note that these pages are for OpenGL 4.5 but most should be equivalent (unless otherwise noted on the function's page).
3. SDL 2 Documentation: [here](#).
4. GLSL Tutorial: [here](#).
5. GLM: [Manual](#) and [Documentation](#).

6. C++ Reference: [here](#)
7. Open Asset Import Library Documentation: [here](#)
8. FreeImage Manual: [here](#).
9. Assignment Framework Documentation: This can be found in "source/doxygen"

## UVs and Textures

How does a particular vertex know which pixel from the texture to use? The answer to that is via UV Mapping ([Wikipedia](#)). Each vertex is given a UV coordinate (usually you do this in your modeling software of choice, see below) which ranges from (0, 0) (bottom left corner) to (1, 1) (top right corner). Then in the vertex/fragment shader, the a texture is sampled from that coordinate to get some values which can be interpreted as anything! For the diffuse and specular textures we choose to interpret the texture values as colors.

To create UV's, you can follow a tutorial for your modeling software. For Maya, a good tutorial can be found [here](#). Look around on Youtube, there are plenty of tutorials for you to follow!

## Assignment

### Assignment 3 Code Explanation

Look in the Assignment3 class documentation in the assignment framework documentation found at "source/-doxygen".

### Provided Asset

You may have noticed that we have provided you a model courtesy of Marcus Dublin on Polycount. You can find the original model thread here: [link](#). This model comes in three parts, the body, backpack, and head/arms which can be found in "assets/outlander/Model". Additionally, this model also comes with textures found in "assets/outlander/TGA". In the assignment 3 source code, you can see how we are using the diffuse and the specular textures. There are also other textures provided, for example: glow, normal, and alpha. For this assignment, it is only mandatory to implement the normal map, but feel free to also incorporate the other textures as well!

### Pointers to Get You Started

- This [link](#) will be very helpful to understand how normal mapping works. The bottom of this link also has some nice links to help you generate normal maps! Additionally the [Wikipedia](#) page may help as well.

### Going Further (Optional)

- Sky boxes are a great way to fill up your background with something rather than a single color. It is what it sounds like. You render a giant box and slap a texture on it. Sky box! You can look at this [tutorial](#) to get started. A more general tutorial for cube mapping can be found [here](#). This [link](#) has some useful links to find skybox textures to download.

## Feedback (Optional)

1. On a scale of 1-10 [10 being the most work]: How much work was this assignment?

## Grading

This assignment will be graded on the following requirements

- There are at least two different geometric models in your scene.
- At least two models have a diffuse texture, specular texture, and a normal texture.
- Normal mapping is implemented correctly.

according to the following rubric.

- + – Exceeds the requirements via one or more artistic/technical contributions
- ✓ – Meets all of the requirements
- – – Does not meet the requirements but still produces a drawing.
- 0 – The submitted solution does not produce a drawing.