**IGB381 Game Engine Technology -** Semester 2 2017

**Report on Global Illumination – Assignment 2**

**For:** Executive Producer ofSpigot **-** Steve Binkman **By:** Michael Cartwright

1. **Description of Work Completed and Work in Progress**

**1.1 Reused Shaders**

The previously developed shaders, FirstPass and SecondPass, have been added to this project to produce Texture Mapping, Normal Mapping, Bump Mapping, Specular Lighting, Diffuse Lighting, Ambient Lighting and Rim Lighting for the terrain and third person character Ethan. Light Attenuation, Multiple lights and Spotlight Cookie Attenuation were also included.

**1.2 Working Techniques Used in New Shaders**

LightandShadowShader, ShadowPass and SplatterMap are the new shader additions for this project.

Based on the feedback provided from the last project, game objects now use a single shader where possible. This shader is the LightandShadowShader. For this project the third player character Ethan and the terrain both utilise this shader. Trees did not have their shaders changed to preserve clean leaves and branches textures.

The ShadowPass shader is designed to be used as a third pass in the LighandShadowShader. This shader uses a shadow casting technique that allow game object’s shadows to be cast directionally based on the position [1]. This was further improved by applying depth bias to the position in the vertex shader [1].

**1.3 Unity Project Work and Techniques Used**

**1.3.1 Terrain Displacement**

At this stage of the report one may consider where any effort was put into this project. Most of the time was spent trying to develop displacement mapped terrain utilising height maps.

A height map of a canyon, taken from [2], was used as the base grounds for the terrain displacement. The height maps used were all converted to grayscale in IBM colour format. The original height map was found to have created lots of sharp edges along the terrain. This prevented Ethan from being able to reliably travel across most of the terrain. Other issues occurred during Unity’s process of the terrain being transformed. To solve both problems the terrain was drawn over in photoshop to smooth out edges. This new height map is currently being used. This technique allowed for the terrain to continue to look realistic and provide slopes at an angle Ethan was capable of walking and jumping over.

**1.4 Techniques to be Developed into New Shaders**

The following techniques were not developed, a work in progress or failed to work.

**1.2.1 Shadows**

Shadows are not received in this project. This includes sampling shadows, which is to identify texture coordinates to allow for other objects to have those shadows appear. The next stage for this was to have multiple shadows to appear based from multiple directional lights. This was to be completed for a directional lightsource(s). If there was time, the SecondPass would have been edited to allow shadows for Spotlight shadows by again sampling the shadow map. From this point soft shadows would have been developed. Here a Shadow Quality option would have been implemented to allow an easy toggling between hard and soft shadows.

**1.2.2 Terrain Displacement**

One technique found during research to apply textures to the terrain was to use a splat map for procedural terrain mapping. Two tutorials and explanations were found online that presented code to allow this to occur.

The first design was to use a C sharp script that analysed the height map data used for the terrain [3]. From here additional work could have been attempted to place into dirt, grass, cliff dirt and cliff rocks along the terrain. A further step would to have those textures blend together.

The second option was to use a shader that blended four textures together and have them applied to a surface [4]. This was the option chosen. However, this feature was not completed. Instead it was pushed assigned to work of shaders for high quality shadows and this report.

**2. The Big Mistake**

It’s at this stage of the report where the reader may question where any time, work or effort was put into this project. Sadly, it’s because I made a terrible mistake. Originally the official IGB381 Global Illumination assessment document was read and understood that “A high quality mesh is used as a basis for a detailed set of hills and valleys generated by a texture height map. A set of high quality textures is used to create realistic terrain appearance. The human figure traverses the terrain realistically.” This was understood by myself as a challenge where I was to make a way to create realistic terrain using a height map and then ensure Ethan can travel along it perfectly. I also took this that the Unity terrain tools could be used to paint textures and trees to make realistic terrain. I then read the Workshop 8, Week 9 workshop and realised I made a massive mistake. One day before this report and project was due.

1. **Description of Old Parameters**

**SunOrbit.cs**

For the SunOrbit.cs file. The parameters available to be tweaked are speed, time and intensity. (Note this script runs on the ThirdPersonCamera in Ethan). Some code and the idea came from Hibberd’s Day/Night Cycle YouTube video [6].

A higher speed value increases how quickly the Sun will orbit around. A negative speed value will result in reversing the direction of the orbit. If speed is 0, the Sun will stand still.

Time reflects where the Sun will be located. The Sun starts at 6:00am to perform a sunrise.

Intensity is designed to increase or decrease the light effect of the directional light. This changes over the day/night cycle.

Sun Trans and Sun are for the Sun directional light source.

textTime is for the User Interface

**Shaders**

These parameters are for all in Game Objects in the scene.

Color Texture allows for a texture\_COLOR to be applied to the Game Object. This also allows for tiling and offset on the x and y planes.

Bump Map allows for a texture\_NRM to be applied to the Game Object. This also allows for tiling and offset on the x and y planes.

Bump Depth allows for a user to modify the bump mapping depth from -2.0 to 2.0.

Color allows for the user to modify the lighting color that appears on the Game Object (With exception to Specular and Rim color).

Specular Color allows for the user to modify the specular color that appears on the Game Object.

Shininess allows the user to adjust how much shine is being applied to the Game Object. A higher value leads to interesting blending effects from two spotlights.

Rim Color allows for the user to modify the rim color effect that appears on the Game Object.

Rim Power allows for the user to modify the rim lighting that appears on the Game Object. A lower value will lead to brighter lighting. A higher value will lead to dimmer lighting.

1. **Source Code Comments**

For custom C# scripts and shader refer to source code in Unity Project Assignment\_2. Commenting explains the process. The following files and their location is shown below.

|  |  |
| --- | --- |
| **File name** | **Location** |
| SunOrbit.cs | Assignment\_2\Assets\Scripts\Lightning |
| EthanShader.cs | Assignment\_2Assets\Objects\Standard  Assets\Characters\ThirdPersonCharacter\Materials |
| FirstPass.cgnic | Assignment\_2\Assets\Shaders |
| SecondPass.cgnic | Assignment\_2\Assets\Shaders |
| ShadowPass.cgnic | Assignment\_2\Assets\Shaders |
| SplatterMap.cgnic | Assignment\_2\Assets\Shaders |
| LightandShadowShader.shader | Assignment\_2\Assets\Shaders |

1. **Statement of Completeness**

Below refers to the criteria from the IGB381 Assignment 1. Below is what has been completed. Everything required should have been implemented.

“SOFTWARE”

* A well designed Object Oriented C# implementation off a 3D scene using Unity scripting

“CONTENT”

* Displacement mapped terrain
* High Quality Shadows

“DOCUMENTATION”

* A description of the shader techniques you used to model the vision, written as a report to Steve Binkman and the game development team
* Source code comments
* Statement of completeness enumerating the assignment sections completed
* References to any content or source code that you used from other sources

1. **References**

[1] J. Flick, "Rendering 7, Shadows, a Unity C# Tutorial", *Catlikecoding.com*, 2017. [Online]. Available: <http://catlikecoding.com/unity/tutorials/rendering/part-7/>. [Accessed: 27- Oct- 2017].

[2] "Grand Canyon Rendering", *Cs.purdue.edu*, 2017. [Online]. Available: <https://www.cs.purdue.edu/homes/ablejde/terrain.html>. [Accessed: 27- Oct- 2017].

[3] "Procedural Terrain Splatmapping", *Alastair Aitchison*, 2017. [Online]. Available: <https://alastaira.wordpress.com/2013/11/14/procedural-terrain-splatmapping/>. [Accessed: 27- Oct- 2017].

[4] J. Flick, "Rendering 3, Combining Textures, a Unity C# Tutorial", *Catlikecoding.com*, 2017. [Online]. Available: <http://catlikecoding.com/unity/tutorials/rendering/part-3/>. [Accessed: 27- Oct- 2017].