Kyles Graphics Report

Kyle Pagan
40209924@live.napier.ac.uk
Edinburgh Napier University - Computer Graphics (SET08116)

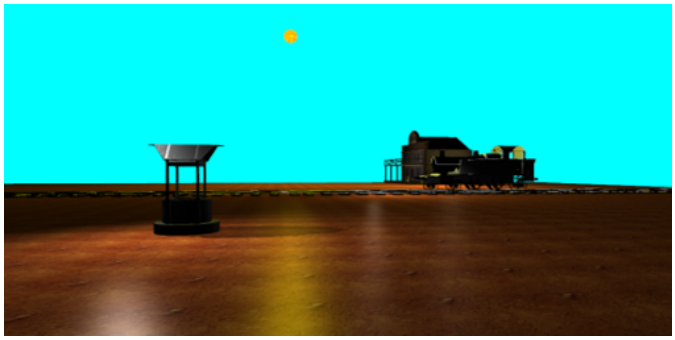

Figure 1: **Full Scene**- Full view of project

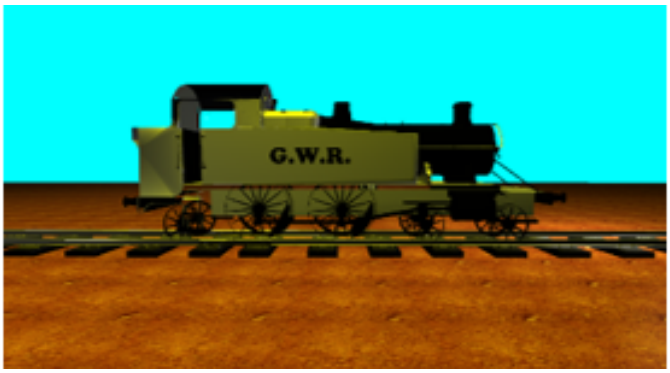

Figure 2: **Train**

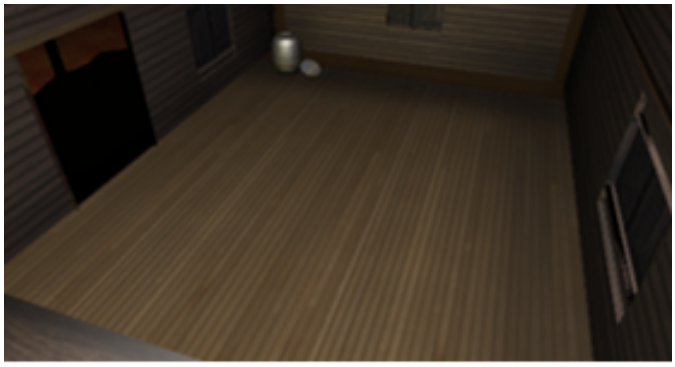

Figure 3: **Saloon** - Inside View

Abstract

This project was created as a combination of OpenGl and C++. The purpose of the project was to demonstrate understanding of texturing, lighting, geometry, post processing and optimisation.

Keywords - textures, lighting, post-processing, graphics, rendering, old west

1 Introduction

The idea for my scene came from the game "Red Dead Redemption" as well as a number of old western movies that I enjoy. I feel that by choosing a scene that I find enjoyable to make I could have more fun with it and I would be able to be more imaginative and creative. Something that seriously appealed to me was the idea of using a steam train in my scene. I feel this gives me a chance to use a few different techniques at one time like a light at the front, the ability to generate steam and transformation (movement). As well as that I was able to make a saloon and be creative with the models I used. There are quite a lot of different effects that this report will cover. 1) The use of 3 lighting effects (ambient, diffuse, and specular) and 3 kinds of lights (point, spot and directional) 2) Textures being used to make the scene look more realistic and colourful. 3) The use of normal mapping to give the illusion that textures have more depth than they do, in my case to make water look like it is wavy. If the meshes normals are not correct then normal mapping will inevitably fail. 4) Shadow mapping, the part of the project I had the most difficulty with, involves using the depth buffer to create a shadow of a mesh using the light hitting the mesh as a reference. It then projects the depth buffers onto the plane. 5) Post-processing, this involves capturing an instance of a render as a texture and then using it in a later instance of a render. The effects that I used were greyscale and masking

2 Related Work

As I mentioned earlier I took inspiration for my scene from the game Red Dead Redemption. This game is based in the old west in 1911. I decided on this because the game is one that I've had countless hours of enjoyment from and I felt passionate about. In addition to this I used to watch a lot of old western time films with my grandpa when I was younger.

3 Implementation

There were a small number of different effects used for this project that were also covered in the introduc- tion.

3.1 Lighting

Adding lighting to the scene was one of the most important aspects. I created 3 spot lights to be used in my scene, one for each mesh. Spot lights are given a a starting position and shine down onto a given area. I also used one point light to shine from my sun to light the en- tire scene, I chose a point light over spot due simply to the fact that a point light will shine in all directions instead of just one like the spot light. To give the lighting effects I used Phong shading by implementing diffuse and specular lighting effects. There is a lot of maths involved in calculating these so I won't go into great detail. The job of specular shading is the show the shiny spot on a surface, if it is too high then even a small light can look as though it is giving of a massive light source. To calculate it involves normalizing the product of the light direction and the viewing direction to calculate the half vector. This is followed by finding the dot product of the half vector and the vertex normal and multiplying them to the power of shininess (this needs to generally be low if the light is small)

3.2 Normal Mapping

Figure 4: **Normal Map** - Illusion of waves in the water

The job of normal mapping is to give a mesh the illusion that it is an uneven surface. To do this it calculates normals on a per pixel instead of per vertex basis. To use normal map coordinates a coordinate space named tangent space is used. To calculate the normal map normal involves creating a 3x3 transformation matrix. Each column takes, in this order, the tangent, binormal and normal vectors.

3.3 Shadows

As I mentioned earlier implementing shadows was the most difficult part of the project. To create a shadow the first render of the scene will calculate where the light is shining from and in what direction it is shining in order to render a shadow map from the point of view of the light. A shadow map is a type of texture that is taken from the

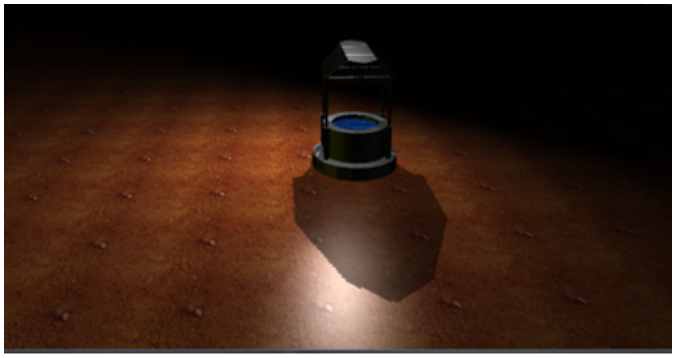

Figure 5: **Shadow** - Shadow cast from spot light to well

depth buffer of the initial render and is then passed on to the following render to deduce what fragments are in the shade factor. This requires the depth space coordinates to be transformed into light space coordinates. In my coursework my shadow is far from being perfect, it is pixelated at the edges and doesn't totally fit the shape of the mesh.

3.4 Post Processing

As mentioned previously post-processing involves capturing a render and using it later on. To do this you need to capture frames using the frame buffer. The frame buffer captures render information as a texture. The steps of using a frame buffer are binding it, rendering the scene, binding back to the screen and then rendering the texture through another shader effect eg, masking and greyscale. To apply the texture to the whole screen a screen quad needs to be created, this is a flat piece of geometry that fills the whole screen, the difference between this and other pieces of the geometry is that transforming the position to screen space isn't necessary, it is already there. Therefore the MVP is an identity matrix. Using this information it is possible to do create many different effects. To create greyscale is very simple. All it involves it calculating the intensity of each pixel, in my project I decided to make mine purple. To use a all that needs are some simple multi-texturing techniques in which a texture is binded to the screen quad

3.5 Optimisation

Optimisation was a very simple process, I used the tools visual studio provided me and found that there were for loops in my program that were making my project run slower, specifically binding the same things, lights, effects etc when rendering my list of meshes, to avoid this I took them out and moved them to before the for loop to save myself from binding needlessly. This the CPU from doing a lot of things it otherwise didn't need to do.

4 Future Work

There is a lot in the part of my coursework that I would like to improve. I plan to work on it a lot over the summer to try and make it better and put myself in better standing

for next semester. I would like to be able to add particles as well as terrain and a skybox.

5 Conclusion

In conclusion I am rather disappointed with the lack of progress I have made between now and the first part of my coursework, there are a number of different things I hoped to have that I don't, as was mentioned earlier. I am pleased with my post processing effects however other than that I have to admit that I am very disappointed in how little I have achieved for part 2