Edinburgh Napier UNIVERSITY

Coursework Report

Emma Parsley
40206111@live.napier.ac.uk
Edinburgh Napier University - Computer graphics (SET08116)

Abstract

This project aim is to create a real time 3d enviroment using OpenGL and C++. The intent is to create a 3D scene shaded as is it where a cartoon. blah blah blah come back to this when there's some stuff in my report.

Keywords – OpenGL, GLFW, Shading, lighting, Non-Photorealistic

1 Introduction

Referencing Leaving this in in case I need to remember how to do this;

You should cite References like this: [1]. The references are saved in an external .bib file, and will automatically be added ot the bibliography at the end once cited.

The project is a scene of three floating islands shown using non-photorealistic rendering to create a visually interesting cartoon-like environment. The project includes use of normal mapping, to show detail on otherwise flat objects, skybox, to create the appearance of an endless environment, multiple lights and sampled Phong shading to cause the cell shaded look.

2 Related Work

3 Implementation

3.1 Lighting

To create a cartoon-like environment the light was calculated using a cell shading technique. A 1D texture of shades which will be sampled from to decide the colour of the light.

Ambient Lighting Ambient lighting is the background lighting which has no direction and so just shows flat colours for the objects in the scene. See Figure 1.

Directional Lighting Directional Lighting effects the whole scene from one direction. In this project it uses the phong shading model;

$$I = I_a K_a + f_{att} I_p (K_d N.L + K_s (R.V)^n)$$

However to create the cell shading effect the diffuse and specular light are clamped and used as texture co-ordinates to sample from a 1D texture of shades then multiplied buy the reflection and light colour. See Figure 2.

Spot and Point Lights The scene can also be lit by spot and point lights which are done in a very similar way to the directional light where we clamp;

$$\frac{max(-R.L,0)^p}{k_c+k_l d+K_q d^2}$$

for spot lights and;

$$\frac{1}{k_c+k_l d+K_q d^2}$$

for point lights to find the texture coordinates to use to sample the shade and multiply it by the light colour and then use the result of that as the light colour for the same calculations we did in directional lighting, including the sampling from the 1D texture again. See figure 3 and figure 4.

Combined Lighting The combined lighting is simply done by adding the calculated colours of each light in the fragment shader. This allows for more than just the shades in the 1D texture to appear making clear the geometry of the object while still restricting the shades enough for it to appear cartoon-like. See figure 5.

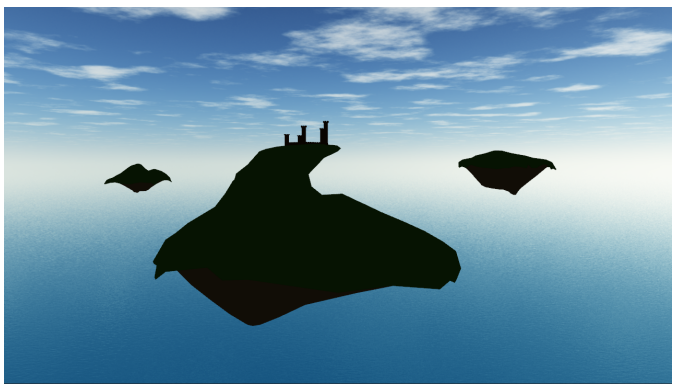

Figure 1: **Ambient Lighting** -Scene with only ambient lighting affecting it

3.2 Normal Mapping

In order to create detail on the flat textures used within the project normal mapping is used. The use of a normal map allows for this detail without needing to render and

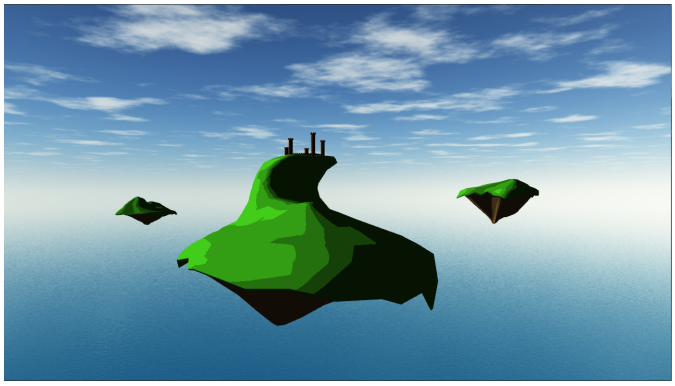

Figure 2: **Directional Lighting** - Scene with only directional lighting affecting it

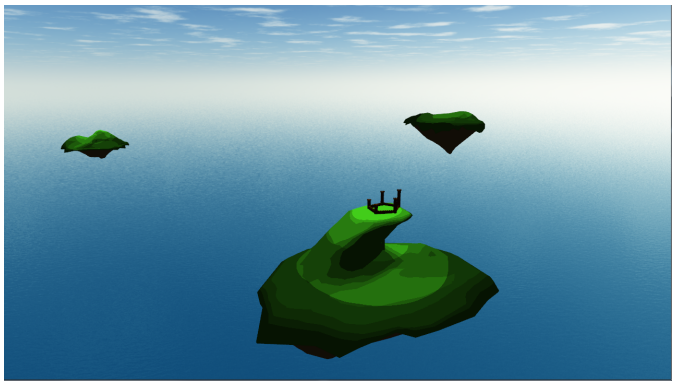

Figure 3: **Spot Lighting** -Scene with only Spot lights affecting it

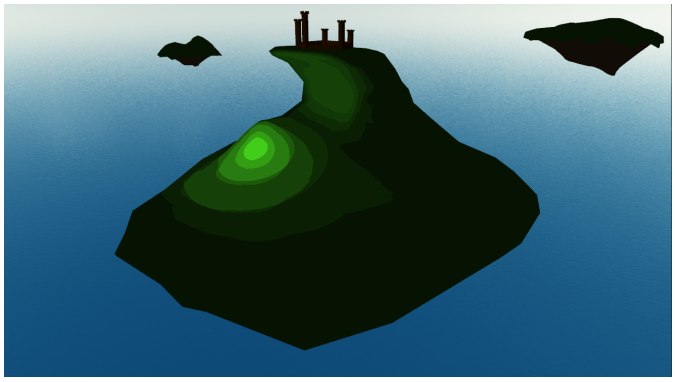

Figure 4: **Point Lighting** - Scene with only Point light affecting it

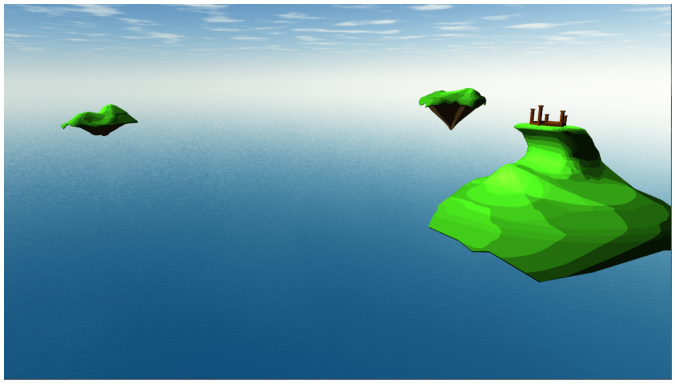

Figure 5: **Multiple Lights** - Scene with all lights affecting it

display more vertices. The castle in figure 6 has just a flat texture on it but when displayed with the normal map it appears to have bricks.

3.3 Skybox

In order to complete the visual pleasing look of this project a skybox was added to create the illusion of an endless environment in the sky above the sea. The skybox is simply a cube scaled up with six seamless textures applied on each face to create the look of the environment. Face culling is disabled on the cube so you can see the textures on the inside, and the cube is moves to the position of the camera so that the user cannot escape the skybox.

4 Future Work

sky Adding clouds to the sky around the islands would add to the overall effect of the scene and visual appeal

Normal maps Adding normal maps to the islands could create a nice grass-like effect over the current flat texture

Outlines Having outlines appear around the objects and over crease edges would really make the objects stand out in the scene and add to the non-photorealistic cartoon style.

5 Formatting

Some common formatting you may need uses these commands for **Bold Text**, *Italics*, and underlined.

5.1 Code Listing

You can load segments of code from a file, or embed them directly.

Listing 1: Hello World! in c++

```
1 #include <iostream>
2
3 int main() {
4     std::cout << "Hello World!" << std::endl;
5     std::cin.get();
6     return 0;
7 }
```

Listing 2: Hello World! in python script

```
1 print "Hello World!"
```

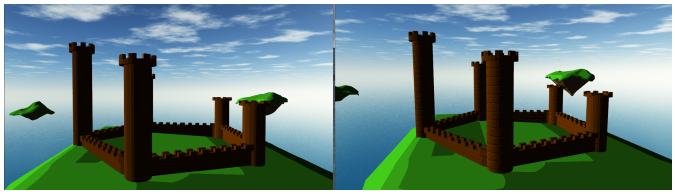

Figure 6: **Normal Mapping** - Left: Castle without normal mapping, Right: castle with normal mapping

6 Conclusion

References

[1] S. Keshav, “How to read a paper,” *SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 83–84, July 2007.