Coursework Report

Sam Reynolds
40205331@live.napier.ac.uk
Edinburgh Napier University - Games Development (SET08116)

Abstract

This report will show the implementation of a graphics coursework. The coursework involves creating a 3D scene using OpenGL and showing the understanding of computer graphics principles.

Keywords – Texturing, Material Shading, Lighting, Cameras, Transforms

1 Introduction

The key effects being shown in the scene are texturing, applied to a skybox, plane and a dragon. Also the use of a point light to brighten up a specific area, in this case directly beneath the dragon.

Adding models to a scene giving them a terrain and a skybox looks visually interesting. Wrapping procedurally generated textures around models gives the dragon a realistic skin like texture and the skybox a custom world, in this case a hell based one, creates an incredibly realistic looking open world.

The difficulties within the project consist of a fair amount of complicated and frustrating maths. The maths involved in the implementation of shader files. As well as the maths, the algorithms implemented are quite technical and take time to fully grip.

Figure 1: **Hell Screenshot** - Screenshot of scene for part1

2 Related Work

The resources used to build the scene mainly came from methods and techniques taught from the graphics workbook. Some extra features such as adding models and textures required searching a little on the internet but mainly came down to manipulating previously developed exercises.

The Idea from the scene came from an interest the dragon Alduin from the skyrim game and the visually cool theme for the skybox fitted well.

Figure 2: **Alduin** - Alduin in the Skyrim game

2.1 LineBreaks

Here is a line

Here is a line followed by a double line break. This line is only one line break down from the above, Notice that latex can ignore this

We can force a break
with the break operator.

2.2 Maths

Embedding Maths is Latex's bread and butter

$$J = \left[\frac{\delta e}{\delta \theta_0}\frac{\delta e}{\delta \theta_1}\frac{\delta e}{\delta \theta_2}\right] = e_{current} - e_{target}$$

2.3 Code Listing

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

Listing 1: Hello World! in c++

```cpp
#include <iostream>

int main() {
    std::cout << "Hello World!" << std::endl;
    std::cin.get();
    return 0;
}
```

Listing 2: Hello World! in python script

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

2.4 PseudoCode

for $i = 0$ **to** 100 **do**
    print_number = true;
    if *i is divisible by 3* **then**
        print "Fizz";
        print_number = false;
    end
    if *i is divisible by 5* **then**
        print "Buzz";
        print_number = false;
    end
    if *print_number* **then**
        print i;
    end
    print a newline;
end

Algorithm 1: FizzBuzz

3 Conclusion

References

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