Coursework Report

Timothy S. Sanderson
40219124@live.napier.ac.uk
Edinburgh Napier University - Computer Graphics (SET08116)

Abstract

This is a project which demonstrates the implementation of basic lighting, shadowing, and texturing effects, in a 3D graphical scene.

Keywords – Graphics, Hierarchy, Phong, Shadows, OpenGL

1 Introduction

This project demonstrates basic, graphical concepts through the use of a 3D environment, rendered using OpenGL. In this scene the main focus is a structure made of numerous moving spheres. The spheres represent a hierarchical relation, as well as providing a good demonstration of the reactivity of the implemented lighting effects. Due to their rapid movement it is easy to see how efficiently the highlights are tracking across their surfaces, and how the shadows are being cast behind them.

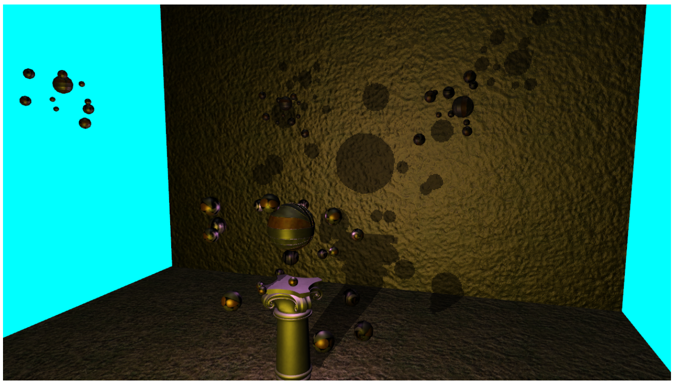

Figure 1: **Scene overview** - A general look at the contents of the scene

2 Implementation

The following are the concepts implemented in the scene to give its current appearance.

2.1 Textures

All objects in this scene are textured. By applying a texture to an object it can be given aesthetic detail that makes it more pleasing to look at, and easier to discern from other objects.

A more complicated use of textures is the normal map. A normal map is used by a shader to manipulate how light shines on an otherwise flat surface. In Figure 1 the back wall is a completely flat plane, but thanks to normal mapping it looks like a roughly textured wall.

2.2 Lighting

Shaders within this scene all implement Phong illumination. This lighting model implements the use of ambient, diffuse, and specular lighting.

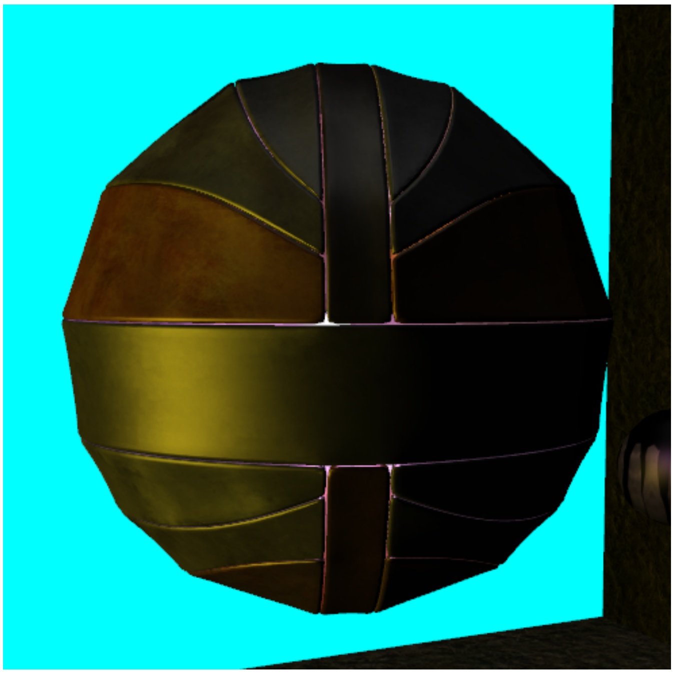

Figure 2: **Lighting** - Lighting effects demonstrated on an orb

On the right of Figure 2 can be seen the ambient light. It's set to be very low, but some details can still be seen.

The areas where details can easily be seen are the ones demonstrating diffuse lighting.

Bright yellow highlights on the left are an example of a specular reflection from the yellow light source.

2.3 Shadows

Shadows are the most complicated aspect of this scene. To cast a shadow, you first have to find how the light views the scene. Once this information has been found, through the use of matrix calculation, you can tell a shader if the light can "see" the vertex that is being rendered. If it is found that the light doesn't have sight of the object, then the brightness of the vertex is reduced. This light reduction creates the effect of shadows.

3 Future Work

A major drawback of this scene is the lack of a skybox. The implementation of a skybox instantly gives the scene a vastly increased aesthetic appeal. In both Figure 1 and 2 there can be seen a bright blue colour in the background. This is a default appearance and not one that looks nice. Hence, a skybox being the next thing that will be added.

The scene could also do with a more cohesive setting/theme, whereas it currently is visually like a tech-demo. Although functional, there is much that could be improved to give the scene more personality while still demonstrating the same concepts. Not only should the scene be more interesting, but it should also be larger. In the current iteration the lights all fight to be seen over the others, and although this does demonstrate the ability to use multiple lights, it makes the scene feel cramped.

4 Conclusion

Overall, this scene has met the requirements made for it. Although it might lack the personality that would make it more interesting to look at, it is still an effective exercise in graphical effect processing.