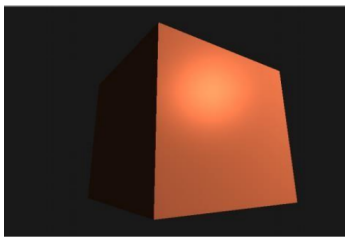
COSC 4370 - Homework 3

Sydney Pospech PSID: 1872314

October 27, 2022

1 Problem

The assignment required the application of our knowledge of OpenGL to practice 3D viewing and by implementing the Phong shader model to create an image as such:



2 Method

The only files that needed code from me were: main.cpp, camera.h, phong.frag, and phong.vs. Within each of these documents I had a specific task to be discussed in the next section; however, I went through the assignment by trying to bring up the red cube before anything else. Once the red cube appeared, I started with the lighting.

3 Implementation

3.1 Main.cpp

The goal for this section was to set up the projection matrix so that the cube will appear in general. This was easily achieved with one line of code declaring the projection matrix setting it equal to the dimensions (camera.Zoom, width/height, 0.1, 100.0). The camera zoom provides the angle in radians. The values 0.1 and 100.0 set the near and the far distances for this orthographic projection.

3.2 Camera.h

In this document we were only asked to calculate the view matrix using Eular angles and the *LookAt* matrix. At first I took a completely inefficient method, and then developed it into a simple one line return statement as follows:

```
return glm::lookAt(Position, Position + Front, Up);
```

3.3 Phong.frag

Going into this document, I started with finishing Phong.vs, because my position needs to be set correctly with glPosition before I could focus on lighting. After that was finished, I went through and calculated ambient, but setting an ambient strength and multiplying it by the lightColor. I then normalized our Normal variable and normalized lightPos - FragPos to become lightDir, the light direction. Then I calculated the diffuse by doing max(dot(norm, lightDir),0.0) * lightColor. To finish it out, I calculated the view direction by normalizing the view position subtracted by the FragPos. I calculated specular by multiplying the strength by pow(max(dot(viewDir, reflectDir), 0,0), 32) * lightColor. I then added all of the lightings ambient, diffuse, and specular and multiplied by the objectColor to obtain our final result.

3.4 Phong.vs

In order to correctly set the position with glPosition, I simply multiplied the projection matrix by the view matrix by the model matrix and then also by vec4(position, 1.0). Not only did I have to set glPosition correctly, but I also had to set FragPos and Normal for Phong.frag to be able to use for lighting. I have to admit that I did take a more unorthodox approach with setting Normal, because I inverted the matrix. I know that this is not the most reliable method; however, it worked for me, so I felt no need in removing it.

4 Results

Overall, I felt like this assignment was relatively simple, due to the guidance that the OpenGL documentation provided me to help me further understand lighting, coordinate systems, color, shading, and cameras.