

## COSC 4370 Homework 3

Vincent Pham 1777469

March 2023

### **Objective**

The goal of the assignment was to create a red cube that was lightened up on one side and darkened on the rest. This was done using Phong shading model, which was provided as part of the assignment.

### **Method**

Most of the program was pre-written for the assignment, using the Phong shading model to create the 3D model. The assignment required the student to write vertex and fragment shaders in the phong.vs file, complete the GetViewMatrix function in the Camera.h file, and the projection matrix in the main.cpp file.

### **Implementation**

#### Project Matrix

The project matrix was created using the perspective command from the glm header file. The instructions on how to implement it was found in a tutorial for how to set up the Camera in OpenGL that will be listed in the sources section.

#### Vertex shader

The vertex shader was created by setting the `gl_Position` equal to the projection multiplied by the model multiplied by the view multiplied by the function of `vec3` using the position. `Vec3 Normal` was set to normal, which was given at the beginning of the file, and `FragPos` was set equal to a function of `vec3` using the model multiplied by the function of `vec4`.

#### Fragment shader

Fragment shaders were created by creating diffused, specular, and ambient lighting. To create diffused lighting, norm and lightDir were normalized and were used to compute the value to diffuse the light at given the color. Specular lighting did the same, but also took into account reflected light and light based on the direction of the camera. Ambient light simply used the ambient strength multiplied by the color of the light multiplied by the color of the object itself. The final result of the fragment shader took into account ambient light, diffused light, and specular light to render the appropriate object color.

## Results

The result of the problem was a red cube that was only bright on one side, and dark on the others since light was simulated to be hitting the front only.

Image:

