COSC 4370 – Homework 3

Name: Grishma Vemireddy

October 2024

1 Objective

In this assignment, we are to implement the 3D viewing and Phong shading model. We should be able to rotate the camera with mouse and shift the camera with W, A, S, D keys. We are to write the vertex and fragment shaders for Phong model to shade a simple cube.

2 Methods

To implement the Phong shader model, there are three types of lighting that must be created: ambient, diffuse, and specular lighting. The combination of these three will result in the Phong shading model. To learn more about the Phong shading model and its implementation, I utilized OpenGL website and the code present there to complete this assignment.
Reference Material: **https://learnopengl.com/Getting-started/Camera** **and** **https://learnopengl.com/Lighting/Basic-Lighting**

3 Implementation

Projection Matrix (main.cpp): Implemented using glm:perspective(glm::radians(camera.Zoom), (float)WIDTH / (float)HEIGHT, 0.1f, 100.0f);

GetViewMatrix() (Camera.h): Implemented this function using the glm::lookAt function that was mentioned in OpenGL section going over Camera/View space. This allows for the movement of the camera based on the user's input.

Vertex shader (phong.vs): Reading in Basic Lighting section in OpenGL, I assign gl_Position to the result of the product of multiplying the position vector with model matrix, view matrix, and the projection matrix. I define Normal to be a normal matrix instead of passing in a normal vector. Normal is set to be the transpose of the inverse of the upper-left 3x3 part of the model matrix. The FragPos (fragment position) is calculated by multiplying the position vector by the model matrix.

Fragment shader (phong.frag): To implement the Phong shading model, three types of lighting are needed: ambient, diffuse, and specular. By reading the documentation on 'Basic Lighting'

section in OpenGL, I implemented my homework code. The ambient light value is set to be a float that represents the ambient strength which is multiplied by light color of the object. Diffuse lighting is achieved by first normalizing the normal vec3 and making a normalized vec3 that's gets the direction from the light source to the fragment position. Then initialize a float diffuse that stores the max of the dot product of the two normalized vectors and zero. Then multiply this value with light color. Secular lighting is based on the reflective properties of surfaces. We initialize a few normalized vec3 variables which hold the view direction, another one which holds the reflect direction and last one calculates light color object with the desired specular strength. All of the three lighting values are added together and multiplied with object color to get the Phong model.

4 Results

These are some screenshots of the resulting cube as the camera is rotated and shifted with mouse and W, A, S, D keys.