

Assignment 1 : Creating your own OpenGL program

NAME: DONGHAO ZHAO

STUDENT NUMBER: 20096438

EMAIL: ZHAODH@SHANGHAITECH.EDU.CN

1 INTRODUCTION

In this homework, I mainly achieved several things as bellow:
1.Thanks to the handy environment provided by TAs, it doesn't bother too much to make the opengl windows.

2.Draw basic triangle and quad then 3D objects on the window, shade the object with the built-in Phong lighting model.

3.Draw a sphere without using built-in functions.

4.Manipulate the cameras to navigate through the virtual scene with the input of keyboard and mouse.

2 IMPLEMENTATION DETAILS

1.Draw plane figure:

We can use `glVertex3f()` to specifies a vertex in 3D. However, we have not enable the lighting effect and the vertex coordinate are on the same plane, so we can think we just draw a plane figure.

2.The algorithm to drawing a sphere:

To draw more complicated sphere, we still start with basic quads. The simplest method to draw a sphere is to use spherical coordinates, and a sphere can be expressed by the following parametric equation:

$$F(u, v) = [\cos(u) \cdot \sin(v) \cdot r, \sin(u) \cdot \sin(v) \cdot r, \cos(v) \cdot r]$$

where: r is the sphere radius; u is the longitude, ranging from 0 to 2π ; and v is the latitude, ranging from 0 to π .

I use two loops for v and u , in first y is set, we can image that we know the latitude then draw a circle along the whole ser latitude, we draw a quad at this point with the help of two tiny move on u and v , so we can have 4 point, then we draw the small quad. To be lively, we draw equator first and pole last.

3.Phong lightening:

OpenGL provides `glLightfv()` to set ambient, diffuse and specular component of the light. According to the guide in the assignment, the light is set properly. It takes some time to adjust the value of `rgb` in sunlight to obtain more obvious light effect.

4.Manipulate the cameras:

I carry out this function by adjust the vector in `gluLookat()`, keyboard input should change the position of the camera, in

other words, walk in my windows, and mouse input should change my view, which means looking at different directions in the world I created.

We have two functions to realize that, `processInput` and `mousecallback`.

Firstly, mouse, the yaw and pitch can be convert to two angles which can be used in changing target position in `gluLookat()`, and the input of the keyboard is more direct, changing the position of camera, which means modify the coordinate of the camera.

In my program, `wasd` and `ol` is used to change position.

3 RESULTS

The first picture is a sphere and a cube with lightening enabled, and the cube is always rotating around the sphere. The second picture is captured when we move the camera position and view to different ones.

