Assignment 1 : Creating your own OpenGL program

NAME: NING BI
STUDENT NUMBER: 8574 5238
EMAIL: BINING@SHANGHAITECH.EDU.CN

1 INTRODUCTION

Basically, I created my first OpenGL program in this assignment. I have finished all the **must** task and also **optional** task. The most methods I used to complete this home work are mainly from *learnOpenGL* website.
The task I have completed are as follow:

(1) Create a basic window-based program for OpenGL rendering, and apply **Multi-sampling** as full-screen anti-aliasing.
(2) Create some basic geometric objects including tetrahedron, cube and spheres, and also shade them with Phong lighting.
(3) Manipulate the objects by translating and rotating them, and manipulate the cameras by constructing the new camera matrix with gluLookAt(...) and use keyboard to control the camera. And I also created a headfile for view changing.
(4) Created a Phong Shading based shader.

2 IMPLEMENTATION DETAILS

Since the assignment itself has already gave us a detail descriptions of whole process, I just briefly describe my method and give some image result.

2.1 Basic rendering and anti-aliasing

During this task, based on basic rendering, I apply **Depth Test** and **Multi Sampling** to achieve a better performance.Here are essential code of depth test and multi sampling:

```
glEnable(GL_DEPTH_TEST);
//enable depth test
glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
// GL_DEPTH_BUFFER_BIT for depth test

glfwWindowHint(GLFW_SAMPLES, 16);
//anti aliasing
//larger parameter leads to more smooth also
  more blur
glEnable(GL_MULTISAMPLE);
//enable anti aliasing
```

2.2 Create basic geometric objects

```
unsigned int VBO, VAO;
glGenVertexArrays(1, &VAO);
glGenBuffers(1, &VBO);
glBindBuffer(...);
glBufferData(...);
glBindVertexArray(VAO);
// position attribute
glVertexAttribPointer(...);
glEnableVertexAttribArray(0);
// normal attribute
glVertexAttribPointer(...);
glEnableVertexAttribArray(1);
...
Shader.use()
// render the cube
glBindVertexArray(VAO);
glDrawArrays(GL_TRIANGLES, start, lines);
```

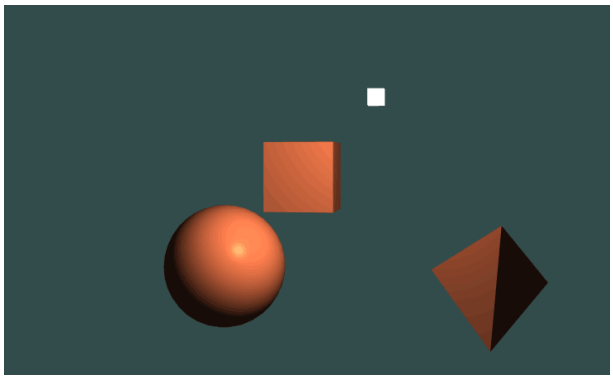

Fig. 1. Phong shading

2.3 Manipulate view

I implement this task in the headfile **'camera.h'**. I could show more details in the demo session.

2.4 Phong shading based shader

Details in **'lightingShader.vs' and 'lightingShader.fs'**.

3 RESULTS

In Figure1 to Figure3 are some output screen shoot of the program.

1:2 • Name: Ning Bi
student number: 8574 5238
email: bining@shanghaitech.edu.cn

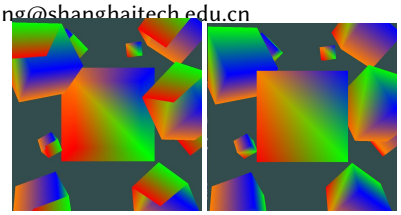

Fig. 2. Depth Test

Fig. 3. Muti Sampling

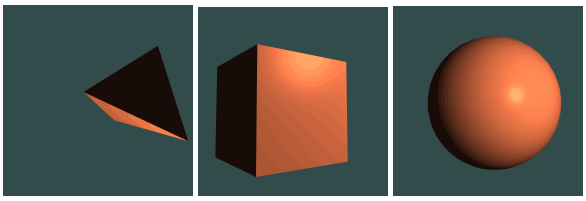

Fig. 4. Geometric objects with Phong Shading