

**Московский авиационный институт**  
**(Национальный исследовательский университет)**  
Факультет информационных технологий и прикладной математики  
Кафедра вычислительной математики и программирования

**Лабораторная работа № 4,5,6**  
по курсу «Компьютерная графика»  
Тема: Использование OpenGL.

Студент: Сорокин Н.Э.  
Группа: М8О-303Б-20  
Преподаватель: Филиппов Г.С.  
Оценка:

Москва, 2022

## 1. Постановка задачи

Ознакомление с технологией OpenGL . Создание шейдерных анимационных эффектов в OpenGL 2.1. Реализация лабораторной работы №3 с использованием OpenGL

## 2. Реализация

```
uniform sampler2D u_texture;
uniform highp vec4 u_lightPosition;
uniform highp float u_lightPower;

varying highp vec4 v_position;
varying highp vec2 v_texcoord;
varying highp vec3 v_normal;

void main(void)
{
    vec4 resultColor = vec4(0, 0, 0, 0);
    vec4 eyePosition = vec4(0, 0, 0, 1);

    vec4 diffMatColor = texture2D(u_texture, v_texcoord);
    vec3 eyeVect = normalize(v_position.xyz - eyePosition.xyz);
    vec3 lightVect = normalize(v_position.xyz - u_lightPosition.xyz);
    vec3 reflectLight = normalize(reflect(lightVect, v_normal));

    float len = length(v_position.xyz - eyePosition.xyz);
    float specularFactor = 100.0;
    float ambientFactor = 0.1;

    vec4 diffColor = diffMatColor * u_lightPower * max(0.0, dot(v_normal, -
lightVect)) / (1.0 + 0.25 * pow(len, 2.0));
    resultColor += diffColor;

    vec4 ambientColor = ambientFactor * diffMatColor;
    resultColor += ambientColor;

    vec4 specularColor = vec4(1, 1, 1, 1) * u_lightPower * pow(max(0.0,
dot(reflectLight, -eyeVect)), specularFactor) / (1.0 + 0.25 * pow(len,
2.0)) ;
    resultColor += specularColor;

    gl_FragColor = resultColor;
}
```

Шейдер (фрагментный) реализующий модель освещения Фонга

```
attribute highp vec4 a_position;
attribute highp vec2 a_texcoord;
attribute highp vec3 a_normal;

uniform highp mat4 u_projectionMatrix;
uniform highp mat4 u_viewMatrix;
uniform highp mat4 u_modelMatrix;

varying highp vec4 v_position;
varying highp vec2 v_texcoord;
varying highp vec3 v_normal;

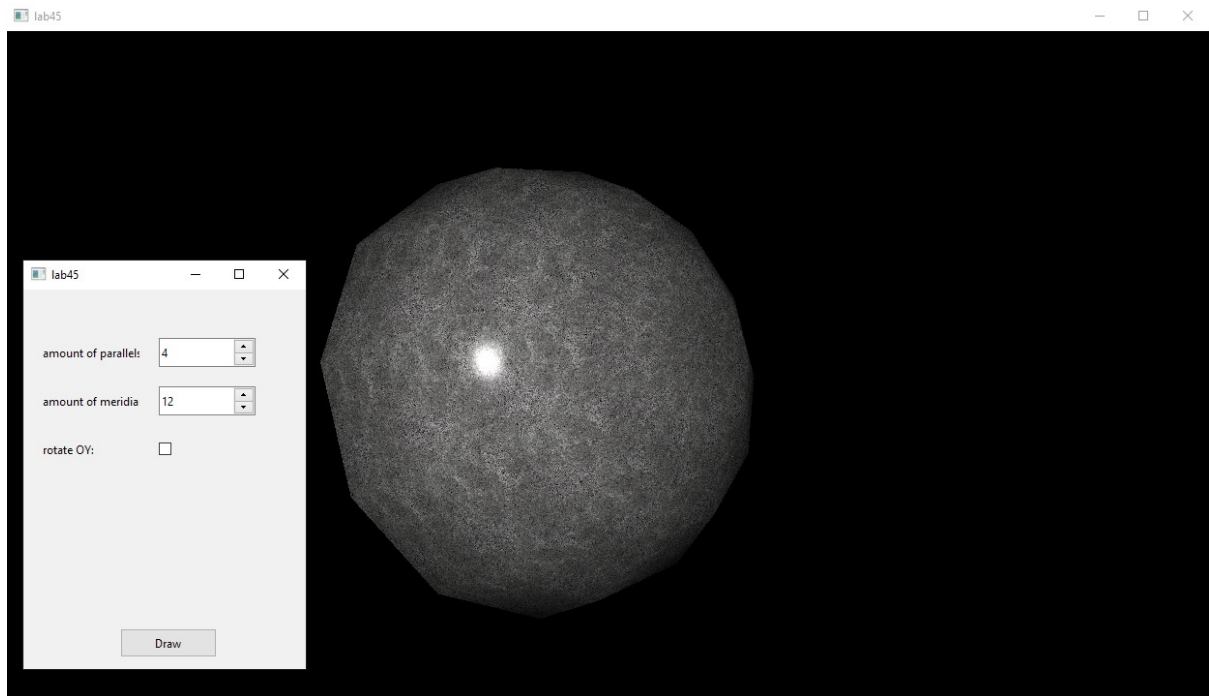
void main(void)
{
    mat4 mv_matrix = u_viewMatrix * u_modelMatrix;

    gl_Position = u_projectionMatrix * mv_matrix * a_position;

    v_texcoord = a_texcoord;
    v_normal = normalize(vec3(mv_matrix * vec4(a_normal, 0.0)));
    v_position = mv_matrix * a_position;
}
```

Вершинный шейдер

### 3. Вывод программы



### 4. Вывод

В ходе данной лабораторной работы я освоил основы работы с инструментарием, предоставляемым QTOpen для реализации модальных диалоговых окон и научился пользоваться процедурой аппроксимации фигур.