

Московский авиационный институт
(Национальный исследовательский университет)

**Институт «Информационных технологий и прикладной
математики»**

Лабораторная работа №4-5
Ознакомление с технологией OpenGL

Работу выполнил:
Рябыкин Алексей Сергеевич
Группа: М8О-309Б-18

Преподаватель: Филиппов Г.С.
Оценка:
Дата:

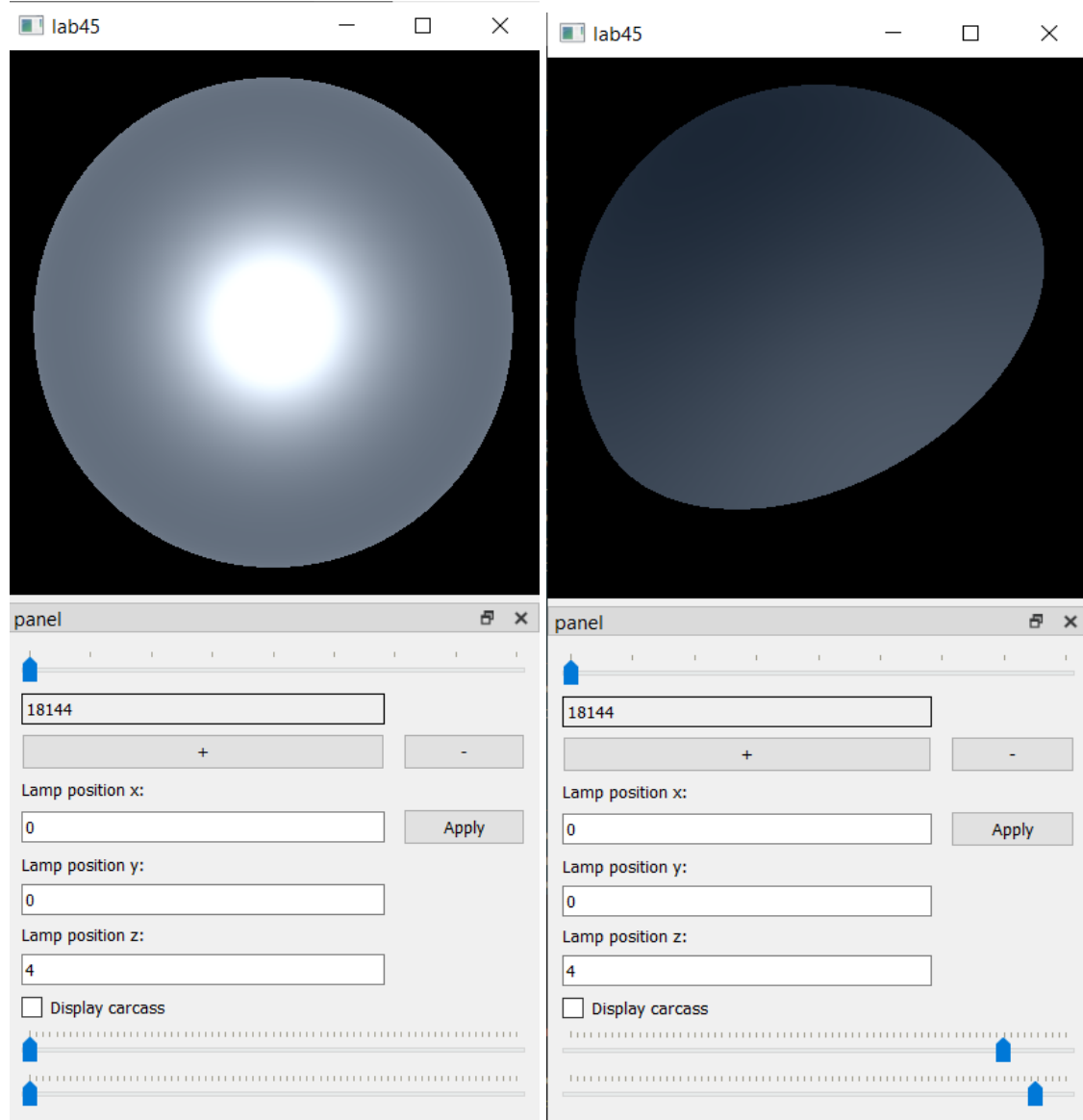
Москва, 2020

Постановка задачи

Создать графическое приложение с использованием OpenGL. Используя результаты Л.Р.№3, изобразить заданное тело (то же, что и в л.р. №3) с использованием средств OpenGL 2.1. Использовать буфер вершин. Точность аппроксимации тела задается пользователем. Обеспечить возможность вращения и масштабирования многогранника и удаление невидимых линий и поверхностей. Реализовать простую модель освещения на GLSL. Параметры освещения и отражающие свойства материала задаются пользователем в диалоговом режиме.

Вариант многогранника: 4. Полушарие

Скриншоты работы программы:



Фрагменты кода:

Функция PaintGL:

```
void View::paintGL() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glEnable(GL_DEPTH_TEST);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    float r;
    r = fig.get_r();
    if (r < 1.f) {
        r = 1.;
    }
    glOrtho(-2. * static_cast<double>(r), 2. * static_cast<double>(r), -2. *
static_cast<double>(r),
        2. * static_cast<double>(r), -2. * static_cast<double>(r), 2. *
static_cast<double>(r));

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    glRotatef(rotateX, 1.f, 0.f, 0.f);
    glRotatef(rotateY, 0.f, 1.f, 0.f);
    glRotatef(rotateZ, 0.f, 0.f, 1.f);
    glScalef(scale, scale, scale);

    if (displayCarcass) {
        glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
        glDisable(GL_LIGHTING);
    } else {
        glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
        glEnable(GL_LIGHTING);
    }

    glPushMatrix();
    glLoadIdentity();
    glEnable(GL_NORMALIZE);
    glMaterialfv(GL_FRONT_AND_BACK, GL_AMBIENT, fig.get_ambient_color());
    glMaterialfv(GL_FRONT_AND_BACK, GL_DIFFUSE, fig.get_diffuse_color());
    glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR, fig.get_specular_color());
    glMaterialf(GL_FRONT_AND_BACK, GL_SHININESS, fig.get_shininess());

    float light_ambient[] = {0.f, 0.22f, 0.51f, 1.f};
    float light_diffuse[] = {1.f, 1.f, 1.f, 1.f};
    float light_specular[] = {1.f, 1.f, 1.f, 1.f};
    float light_position[] = {lightPositionX,
                             lightPositionY,
                             lightPositionZ, 1.f};

    glEnable(GL_LIGHT0);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
    glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
    glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);
    glLightf(GL_LIGHT0, GL_SPOT_EXPONENT, 128);
    glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, 1.f);
    glPopMatrix();
    glEnable(GL_CULL_FACE);
    glCullFace(GL_BACK);
    glColor3f(1.f, 0.f, 0.f);
    for (auto polygon: fig.get_polygons()) {
        glBegin(GL_POLYGON);
        for (auto vertex: polygon.vertices) {
            glVertex3f(vertex.x(), vertex.y(), vertex.z());
        }
        glEnd();
    }
    glDisable(GL_CULL_FACE);
    glDisable(GL_LIGHT0);
    glDisable(GL_LIGHTING);
    glDisable(GL_DEPTH_TEST);
}
```

Среда разработки: Qt Creator 4.10.1

Вывод: В процессе выполнения лабораторной работы научился отрисовывать, масштабировать, центрировать при изменении окна, вращать и удалять невидимые линии для отрисовки выпуклых тел для одного источника света с использованием OpenGL.