**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

**Лабораторная работа № 4-5**

Тема: Ознакомление с технологией OpenGL

Студент: Айрапетова Евгения Ашотовна

Группа: 80-306

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2021

1. Постановка задачи

Создать графическое приложение с использованием OpenGL. Используя результаты ЛР3, изобразить заданное тело (то же, что и в л.р. 3) с использованием средств OpenGL 2.1 (или выше) . 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.

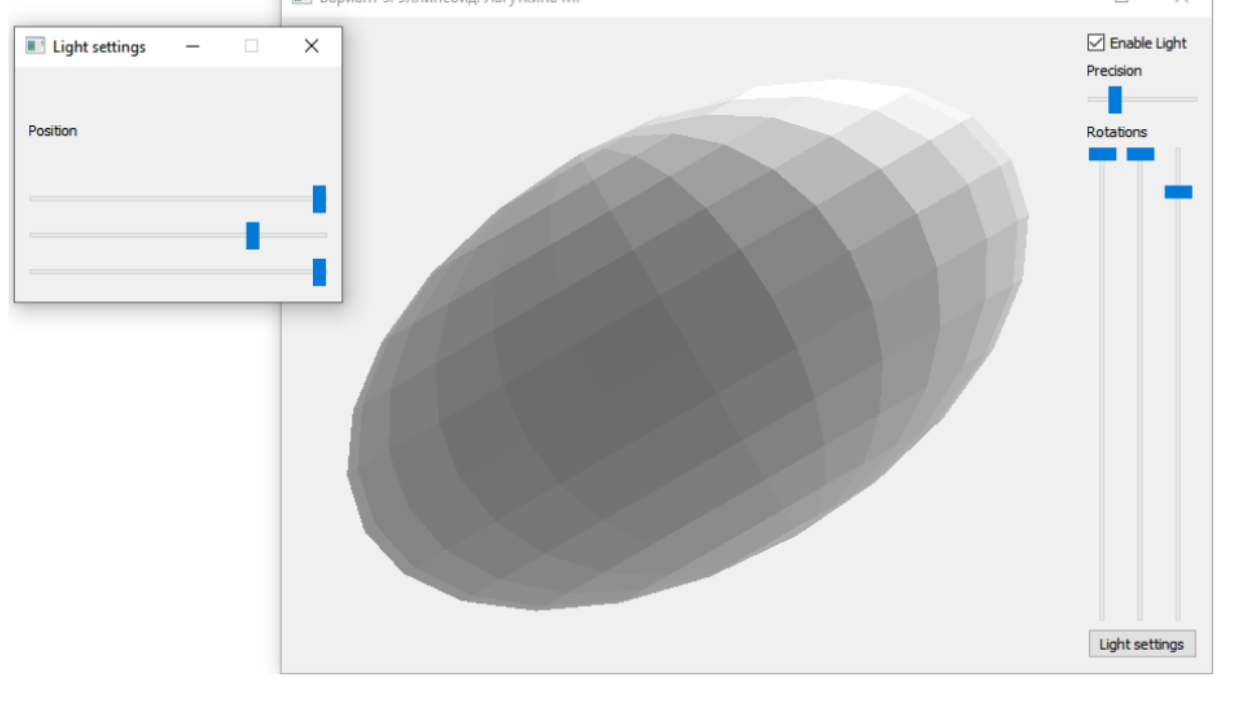
1. Описание программы

Для решения задачи были реализованы классы для вычисления вершин эллипсоида, класс для отображения освещения и самого виджета.

Изменение параметров эллипсоида реализовано в классе ViewWidget. Можно изменить параметр аппроксимации (бегунок Precision), а также выбрать режим просмотра: каркасный или режим поверхности (чек-бокс Enable Light). Также во втором режиме доступно изменение освещения фигуры.

Изменение параметров освещения описано в классе LightSettingsWidget. Можно передвигать источник освещения по X, Y, Z. В диалоговом окне изменение производится засчет изменения значений на трех бегунках соответственно.

1. Результаты выполнения тестов



1. Листинг программы

glwidget.h

#ifndef GLWIDGET\_H

#define GLWIDGET\_H

#define GL\_GLEXT\_PROTOTYPES

#include<QtOpenGL>

#include"glfigureengine.h"

class GlWidget : public QGLWidget {

public:

GLFigureEngine Engine;

GlWidget(QWidget\* parent = nullptr);

void initializeGL() override;

void resizeGL(int nWidth, int nHeight) override;

void paintGL() override;

void scene();

void read(int a);

void cuboid();

};

#endif

viewvidget.h

#ifndef VIEWIDGET\_H

#define VIEWIDGET\_H

#include <QWidget>

#include <QHBoxLayout>

#include <QVBoxLayout>

#include <QCheckBox>

#include <QSlider>

#include <QPushButton>

#include "glwidget.h"

#include "glellipsoid.h"

#include "lightsettingswidget.h"

const int MAX\_VALUE\_SLIDER = 1000;

class ViewWidget : public QWidget

{

Q\_OBJECT

private:

QHBoxLayout\* hLayout = new QHBoxLayout(this);

QVBoxLayout\* vLayout = new QVBoxLayout();

QVBoxLayout\* vLayoutView = new QVBoxLayout();

QHBoxLayout\* hLayoutFigures = new QHBoxLayout();

QHBoxLayout\* hLayoutRotations = new QHBoxLayout();

public slots:

void configChanged(int);

void configChanged();

void showLightSettings();

public:

LightSettingsWidget\* lsw = new LightSettingsWidget();

ViewWidget();

~ViewWidget();

GlWidget\* glw = new GlWidget;

QCheckBox\* isIzomteric = new QCheckBox();

QCheckBox\* isHidingBack = new QCheckBox();

QCheckBox\* isLigthing = new QCheckBox();

QSlider\* sliderRotX = new QSlider(Qt::Vertical);

QSlider\* sliderRotY = new QSlider(Qt::Vertical);

QSlider\* sliderRotZ = new QSlider(Qt::Vertical);

QSlider\* sliderPrecision = new QSlider(Qt::Horizontal);

QPushButton\* ligthSettings = new QPushButton();

GLEllipsoid\* sph = new GLEllipsoid(0.7,100, 1.7,1,1);

};

#endif

lightsettingswidget.h

#ifndef LIGHTSETTINGSWIDGET\_H

#define LIGHTSETTINGSWIDGET\_H

#include <QWidget>

#include <QSlider>

#include <QVBoxLayout>

#include <QHBoxLayout>

#include <QPushButton>

#include <QColorDialog>

const int MAX\_VALUE\_SLIDER\_LW = 1000;

class LightSettingsWidget : public QWidget

{

Q\_OBJECT

QVBoxLayout\* vLayout = new QVBoxLayout(this);

QHBoxLayout\* hLayout = new QHBoxLayout();

bool isAmbient = false;

bool isSpecular = false;

bool isDiffuse = false;

public:

QColor ambient = Qt::white;

QColor specular = Qt::white;

QColor diffuse = Qt::white;

QColorDialog\* cD = new QColorDialog();

QSlider\* sliderPositionX = new QSlider(Qt::Horizontal);

QSlider\* sliderPositionY = new QSlider(Qt::Horizontal);

QSlider\* sliderPositionZ = new QSlider(Qt::Horizontal);

QSlider\* sliderPositionW = new QSlider(Qt::Horizontal);

LightSettingsWidget();

public slots:

void getColorDialogA();

void getColorDialogS();

void getColorDialogD();

};

#endif

gllight.h

#ifndef GLLIGHT\_H

#define GLLIGHT\_H

#include <QColor>

#include <QVector4D>

class GLLight

{

public:

QColor diffuse = Qt::blue;

QColor specular = Qt::blue;

QColor ambient = Qt::blue;

QVector4D direction = {0.0,0.0,0.0,1.0};

GLLight();

};

#endif

glfigureengine.h

#ifndef GLFIGUREENGINE\_H

#define GLFIGUREENGINE\_H

#include<QVector>

#include<QQueue>

#include<QtOpenGL>

#include "glabstractpainting.h"

#include "gllight.h"

class GLFigureEngine {

private:

float rotateX = 0.0;

float rotateY = 0.0;

float rotateZ = 0.0;

float scale = 1.0;

bool isIzomteric = false;

bool isOrtho = true;

bool isHiddingBack = false;

bool isLightEnab = false;

QVector<GLAbstractPainting\*> paintings;

QQueue<GLenum> enableMods;

QQueue<GLenum> disableMods;

void applyTransform() const;

void applyLight() const;

void enable();

void disable();

void disableBooleanProj();

void prePaintSettings(GLAbstractPainting\* figure);

public:

GLFigureEngine();

GLLight light;

void paint();

void addFigure(GLAbstractPainting\* f);

void deleteFigures();

~GLFigureEngine();

void setRotation(float x, float y, float z);

float getRotationX() const;

float getRotationY() const;

float getRotationZ() const;

void setRotationX(float a);

void setRotationY(float a);

void setRotationZ(float a);

void Enable(GLenum m);

void Disable(GLenum m);

void setOrthoProjection();

bool isOrthoProjection();

bool isHidingBack() const;

void setHidingBack(bool b);

void setProjection(int w, int h);

bool isLightEnabled() const;

void setLightEnabled(bool b);

};

#endif

glellipsoid.h

#ifndef GLELLIPSOID\_H

#define GLELLIPSOID\_H

#include "glabstractpainting.h"

class GLEllipsoid : public GLAbstractPainting

{

float radius, precision, a, b, c;

public:

GLEllipsoid(float r, float p,float A, float B,float C);

float getRadius() const;

void setRadius(float r);

float getPrecision() const;

void setPrecision(float p);

void setABC(float A, float B,float C);

void paint() override;

};

#endif

main.cpp

//Айрапетова

#include "glwidget.h"

#include <QApplication>

#include <viewwidget.h>

int main(int argc, char \*argv[]) {

QApplication a(argc, argv);

ViewWidget\* view = new ViewWidget;

view->setWindowTitle("Эллипсоид");

view->show();

return a.exec();

}

ВЫВОД

В процессе выполнения данной ЛР я работала с телами в OpenGl, аппроксимацией и освещением.

ЛИТЕРАТУРА

Документация по Qt: https://doc.qt.io (дата обращения 05.12.2021).