**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Компьютерная графика»**

Тема:**Примитивы OpenGL**

|  |  |  |
| --- | --- | --- |
| Студент гр. 8382 |  | Мирончик П.Д. |
| Преподаватель |  | Герасимова Т.В. |

Санкт-Петербург

2020

# Задание

Разработать программу, реализующую представление определенного набора примитивов из имеющихся в библиотеке OpenGL (GL\_POINT, GL\_LINES, GL\_LINE\_STRIP, GL\_LINE\_LOOP, GL\_TRIANGLES, GL\_TRIANGLE\_STRIP, GL\_TRIANGLE\_FAN, GL\_QUADS, GL\_QUAD\_STRIP, GL\_POLYGON).

Разработанная на базе шаблона программа должна быть пополнена возможностями остановки интерактивно различных атрибутов примитивов рисования через вызов соответствующих элементов интерфейса пользователя

# Ход работы

Для разработки приложения использовалась IDE Qt Creator с целью облегчения использования модулей фреймворка Qt (в том числе настройки, подсказки, графический редактор и т.п.).

Для настройки проекта достаточно создать приложение Qt Widgets и добавить в .pro файл

## QT += core gui opengl // подключение модуля opengl

## …

## LIBS += -lopengl32 -lglu32 // добавление библиотек opengl

MainWindow - класс разметки. Окно, в котором находятся компоненты управления и виджет для работы с opengl.

Блок управления (справа) задается через графический интерфейс QtDesigner. После инициализации ui->setupUi(this) производится добавление кастомного OpenGLWidget, настройка весов элементов для более аккуратного отображения и задаются позиции выпадающего списка режимов.

OpenGLWidget - работа с opengl. Слот onModeChanged подписывается на изменения позиции выпадающего списка и вызывает перерисовку виджета при изменении позиции (при изменении размеров окна перерисовка вызывается автоматически). Для отрисовки используются 2 функции:

paintGL - вызывается родительским классом. Задает матрицу проецирования 20 на 20 по x и y, изменяя ориентацию экрана (оси теперь идут сверху-вниз слева-направо), очищает экран и рисует требуемые фигуры:

## glClearColor(0, 0, 0, 1);

## glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

## glMatrixMode(GL\_PROJECTION);

## glLoadIdentity();

## glOrtho(0,20,20,0,1,0);

## glBegin(MODES[\_mode]);

## paintVertices();

## glEnd();

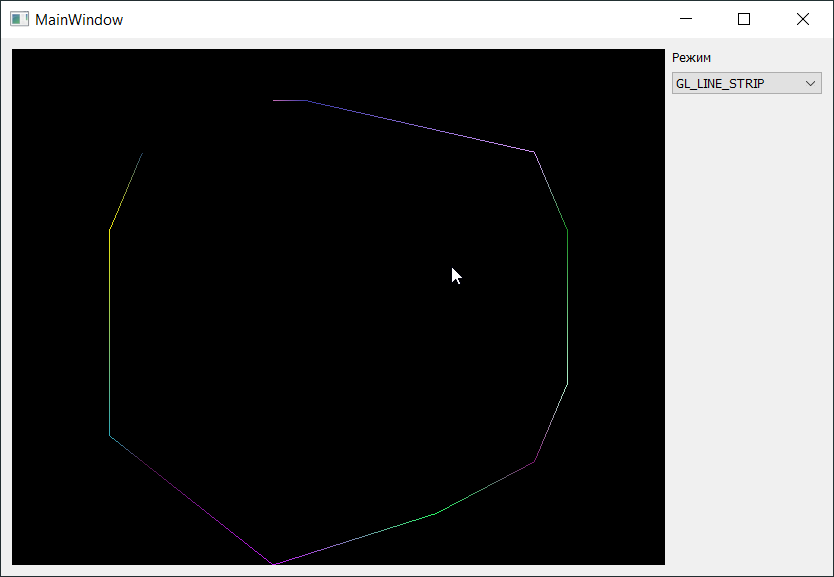
paintVertices - задает список вершин и их цветов для фигур (используется 12 вершин, потому что такой набор подойдет как для GL\_TRIANGLES, где требуется n % 3 вершин, так и для GL\_QUADS с n % 4).

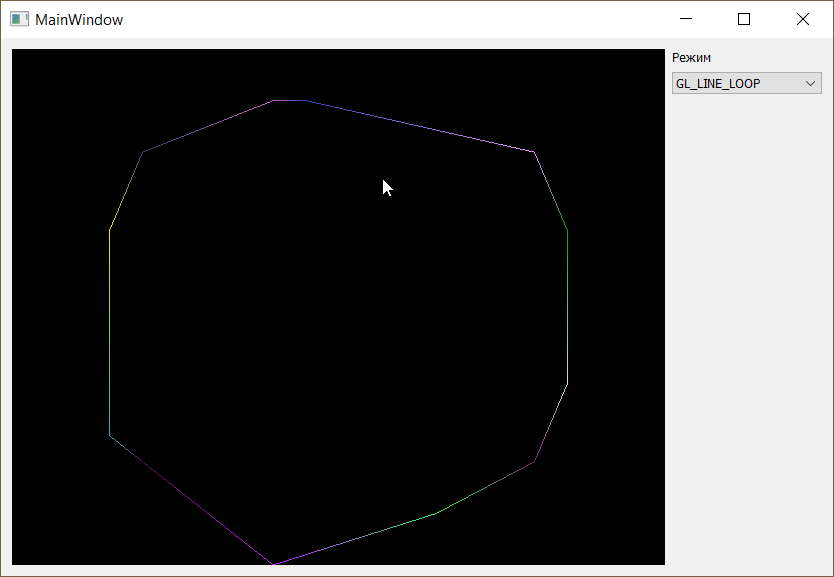
Ссылка на исходный код: <https://github.com/Pavel-by/sem6_ComputerGraphics/tree/master/lr1/lr1>

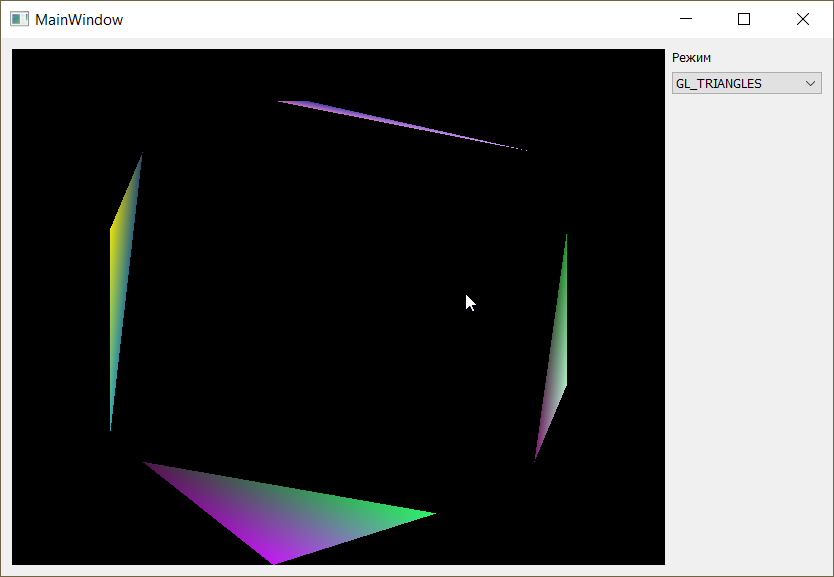
# Тестирование

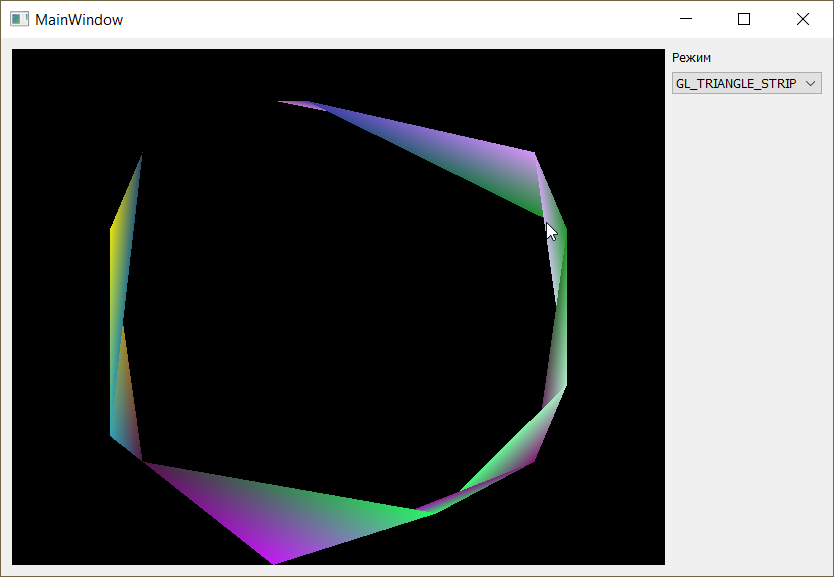
Ссылка на демонстрацию: <https://github.com/Pavel-by/sem6_ComputerGraphics/raw/master/lr1/demonstration.mp4>

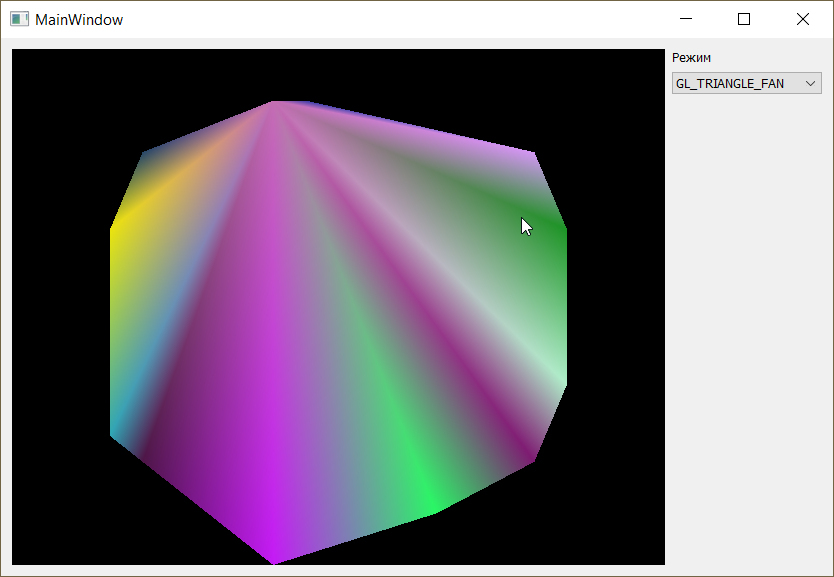


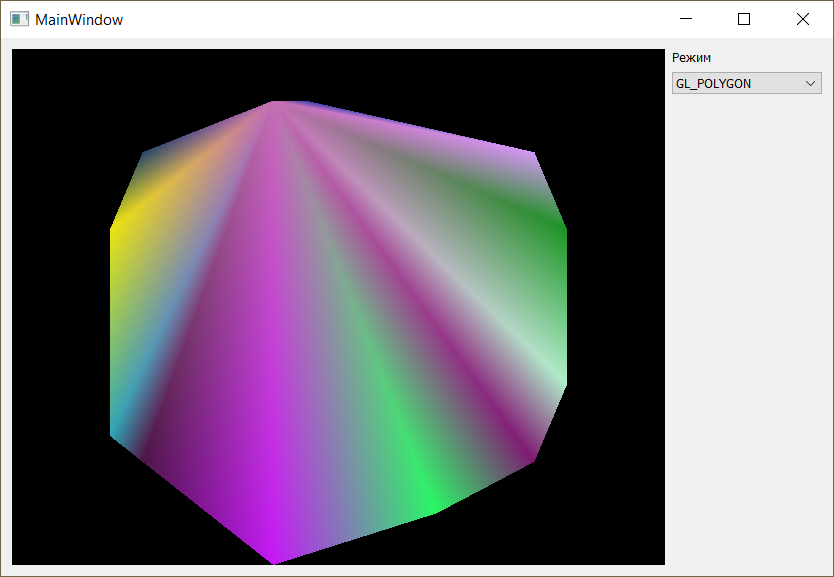


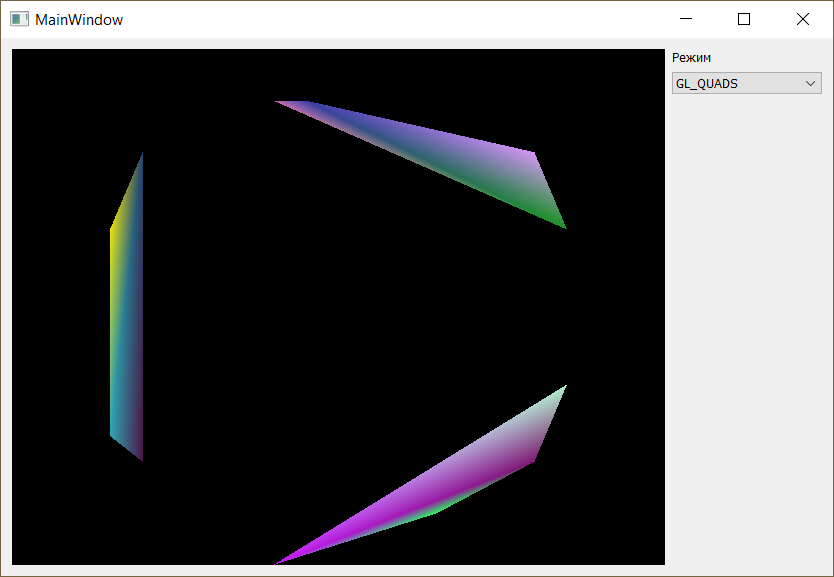


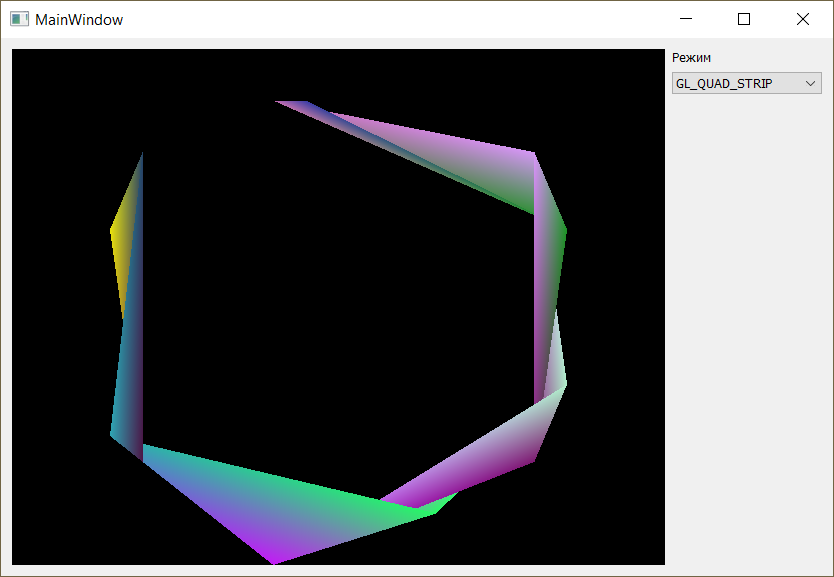












# Вывод

В результате выполнения лабораторной работы была разработана программа, создающая графические примитивы OpenGL. Программа работает корректно. При выполнении работы были приобретены навыки работы с графической библиотекой OpenGL.