МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ

по лабораторной работе №1
по дисциплине «Компьютерная графика»
Тема: «Примитивы OpenGL»

Студентка гр. 7381 _____ Алясова А.Н.

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

Общие сведения.

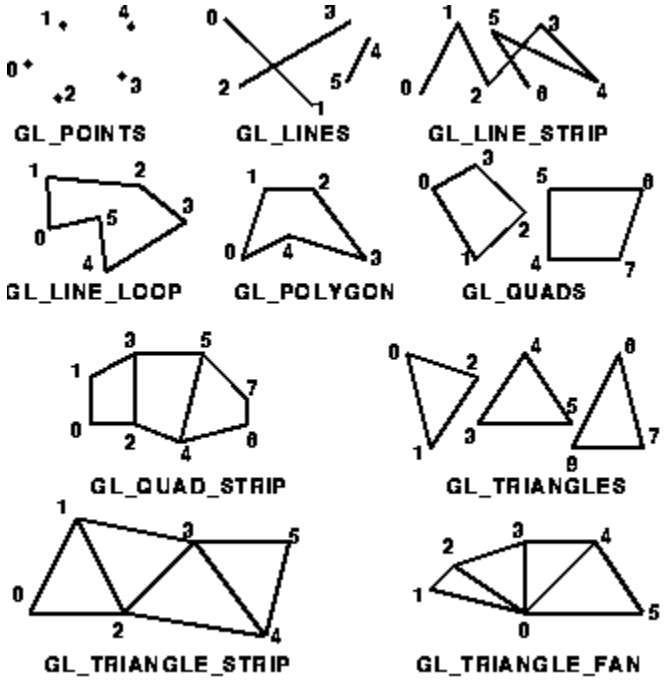
В данной лабораторной работе должны быть рассмотрены следующие примитивы:

1) ***GL_POINTS*** – каждая вершина рассматривается как отдельная точка, параметры которой не зависят от параметров остальных заданных точек. При этом вершина n определяет точку n. Рисуется N точек (n – номер текущей вершины, N – общее число вершин).
2) ***GL_LINES*** – каждая пара вершин рассматривается как независимый отрезок. Первые две вершины определяют первый отрезок, следующие две – второй отрезок и т.д., вершины (2n-1) и 2n определяют отрезок n. Всего рисуется N/2 линий. Если число вершин нечетно, то последняя просто игнорируется.
3) ***GL_LINE_STRIP*** – в этом режиме рисуется последовательность из одного или нескольких связанных отрезков. Первая вершина задает начало первого отрезка, а вторая – конец первого, который является также началом второго. В общем случае, вершина n (n > 1) определяет начало отрезка n и конец отрезка (n - 1). Всего рисуется (N - 1) отрезок.

4) ***GL_LINE_LOOP*** – осуществляется рисование замкнутой кривой линии. Первая вершина задает начало первого отрезка, а вторая – конец первого, который является также началом второго. В общем случае, вершина n (n > 1) определяет начало отрезка n и конец отрезка (n - 1). Первая вершина является концом последнего отрезка. Всего рисуется N отрезков.
5) ***GL_TRIANGLES*** – каждая тройка вершин рассматривается как независимый треугольник. Вершины (3n-2), (3n-1), 3n (в таком порядке) определяют треугольник n. Если число вершин не кратно 3, то оставшиеся (одна или две) вершины игнорируются. Всего рисуется N/3 треугольника.
6) ***GL_TRIANGLE_STRIP*** - в этом режиме рисуется группа связанных треугольников, имеющих общую грань. Первые три вершины определяют первый треугольник, вторая, третья и четвертая – второй и т.д. для нечетного n вершины n, (n+1) и (n+2) определяют треугольник n. Для четного n треугольник определяют вершины (n+1), n и (n+2). Всего рисуется (N-2) треугольника.
7) ***GL_TRIANGLE_FAN*** - в этом режиме рисуется группа связанных треугольников, имеющих общие грани и одну общую вершину. Первые три вершины определяют первый треугольник, первая, третья и четвертая – второй и т.д. Всего рисуется (N-2) треугольника.
8) ***GL_QUADS*** – каждая группа из четырех вершин рассматривается как независимый четырехугольник. Вершины (4n-3), (4n-2), (4n-1) и 4n определяют четырехугольник n. Если число вершин не кратно 4, то оставшиеся (одна, две или три) вершины игнорируются. Всего рисуется N/4 четырехугольника.

9) ***GL_QUAD_STRIP*** – рисуется группа четырехугольников, имеющих общую грань. Первая группа из четырех вершин задает первый четырехугольник. Третья, четвертая, пятая и шестая задают второй четырехугольник.

10) ***GL_POLYGON*** **–** задет многоугольник. При этом число вершин равно числу вершин рисуемого многоугольника.



Ход работы.

Работа выполнена с использованием IDE Qt на языке программирования C++. Применены графические возможности библиотеки Qt для создания виджетов, таких как: выпадающий список.

Для того чтобы настроить среду для совместной работы Qt и OpenGl необходимо скачать библиотеку OpenGl и настроить Qt для работы с последней: произвести изменения в .pro файле:

```
QT       += core gui opengl
```

Для создания окна, в котором будет происходить отображение примитивов OpenGl создан класс-наследник от QWidget. В этот класс добавлен выпадающий список (QComboBox) для выбора в нем необходимого для отрисовки примитива.

При выборе значения из выпадающего списка, с помощью механизма сигналов и слотов вызывается функция, которая задает значение переменной с помощью switch-case.

Реализован класс glView являющийся наследником QGLWidget, в этом классе для отрисовки примитивов OpenGl необходимо реализовать 3 виртуальных метода класса-родителя, а именно:

`void initializeGL();` - инициализация OpenGl

`void paintGL();` – непосредственно рисование примитивов

`void resizeGL(int w, int h);` – настройка с учетом размера экрана

Результаты выполнения программы.

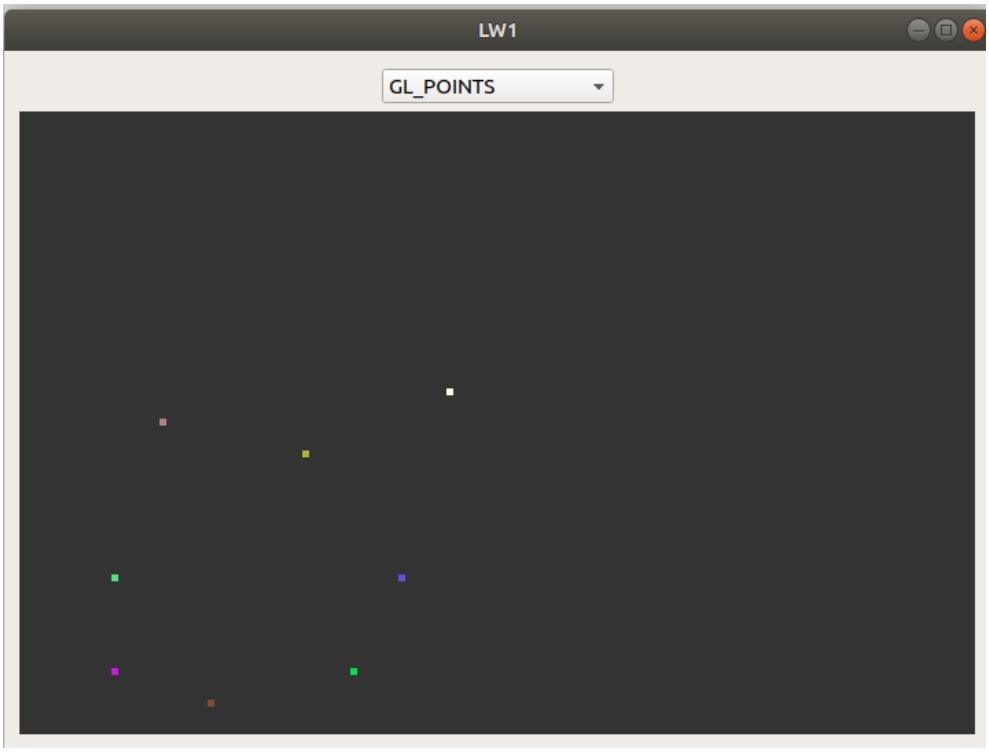


Рисунок 1 – Примитив GL_POINT

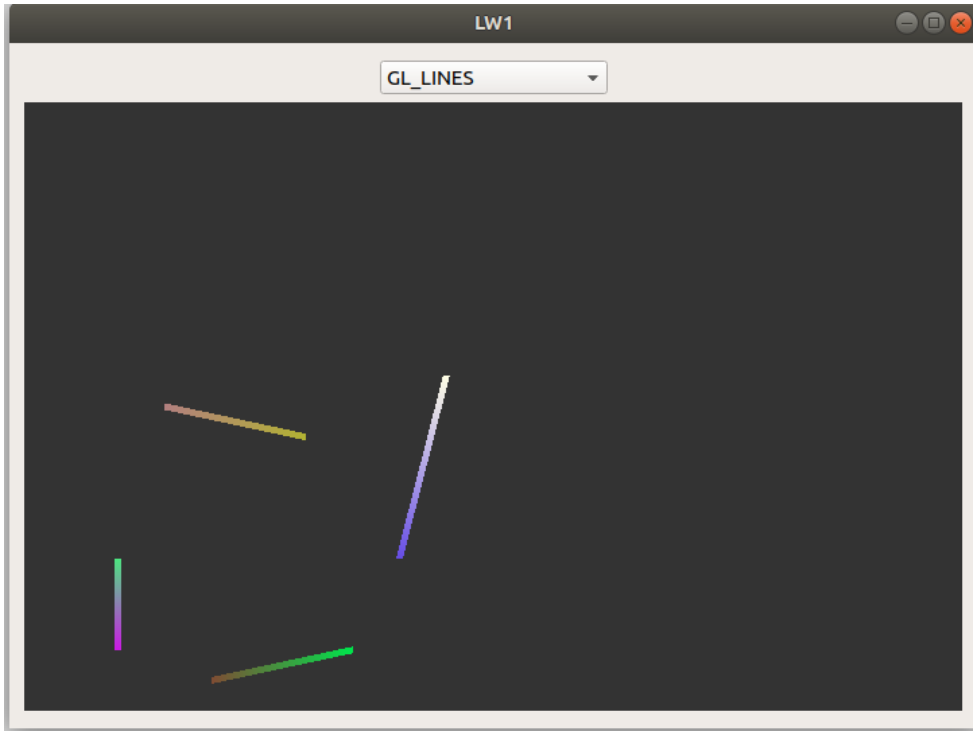


Рисунок 2 – Примитив GL_LINES

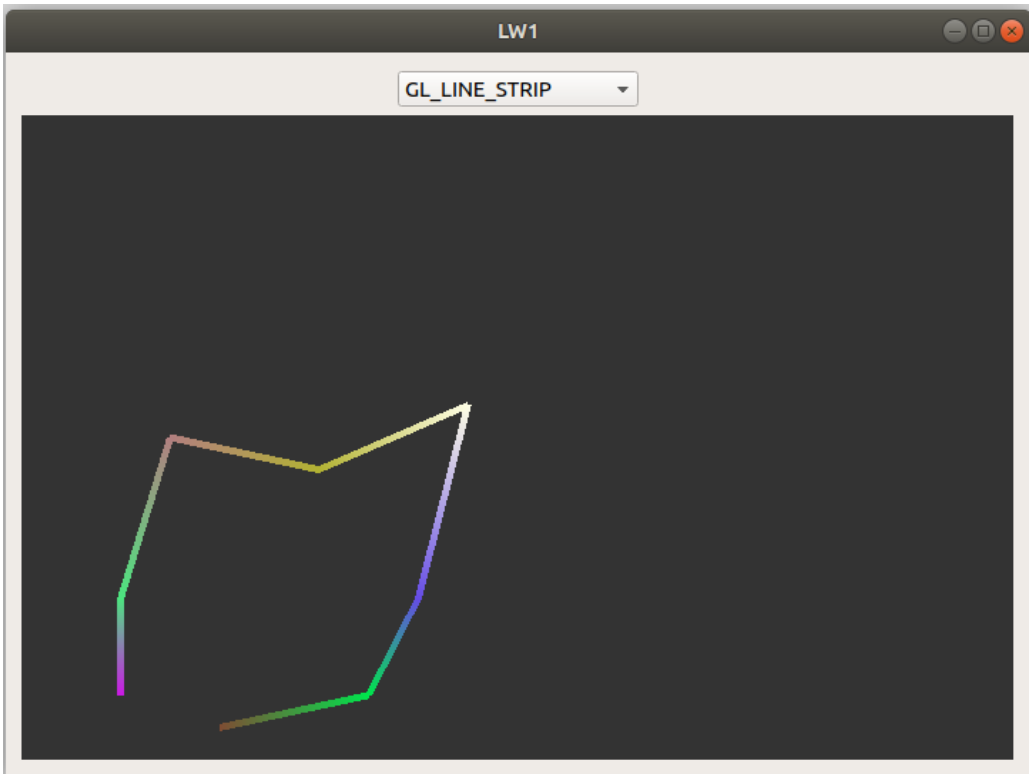


Рисунок 3 – Примитив GL_LINE_STRIP

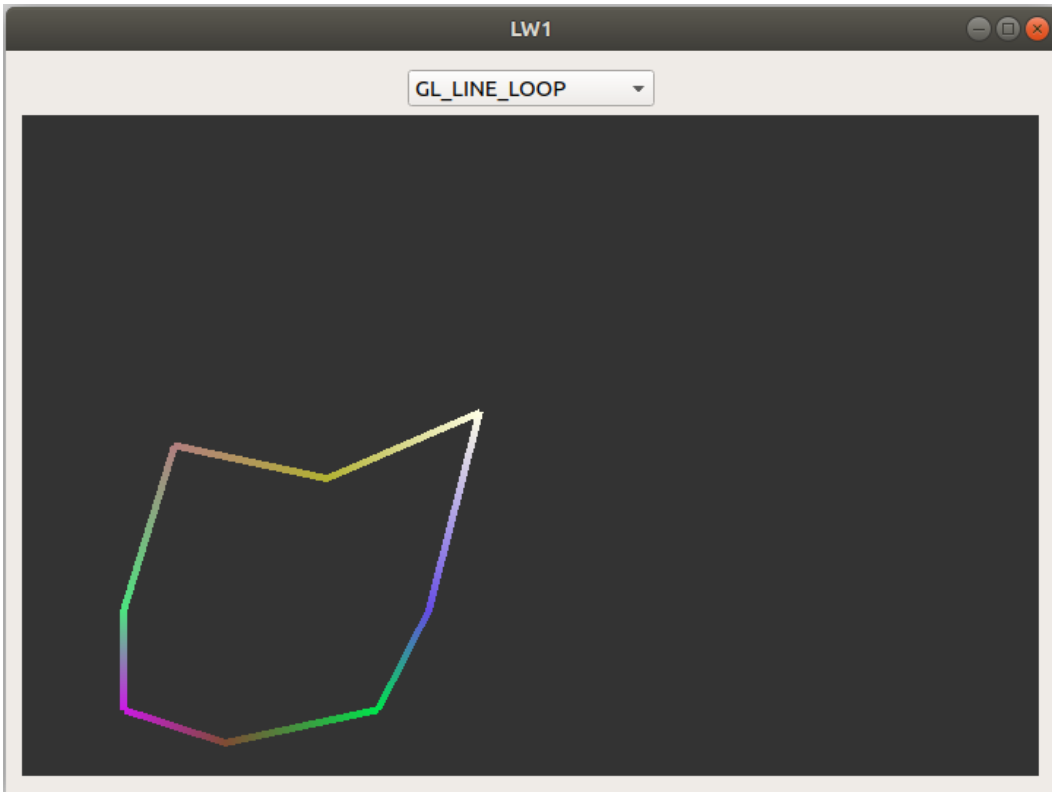


Рисунок 4 – Примитив GL_LINE_LOOP

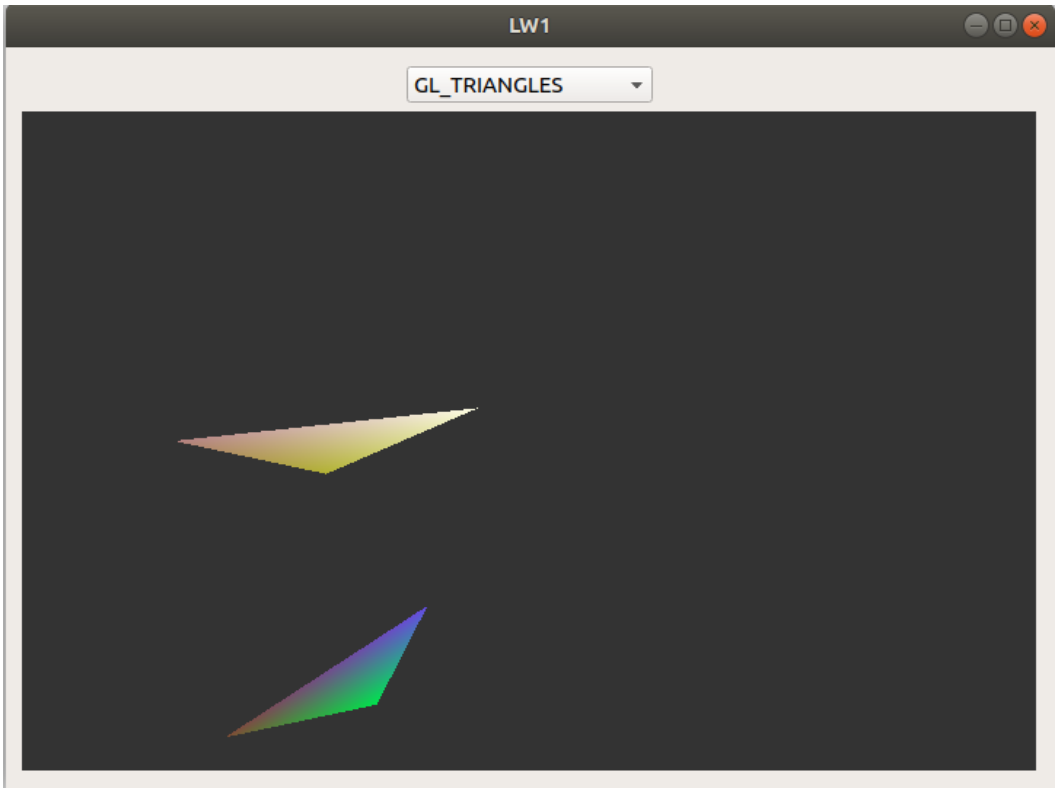


Рисунок 5 – Примитив GL_TRIANGLES

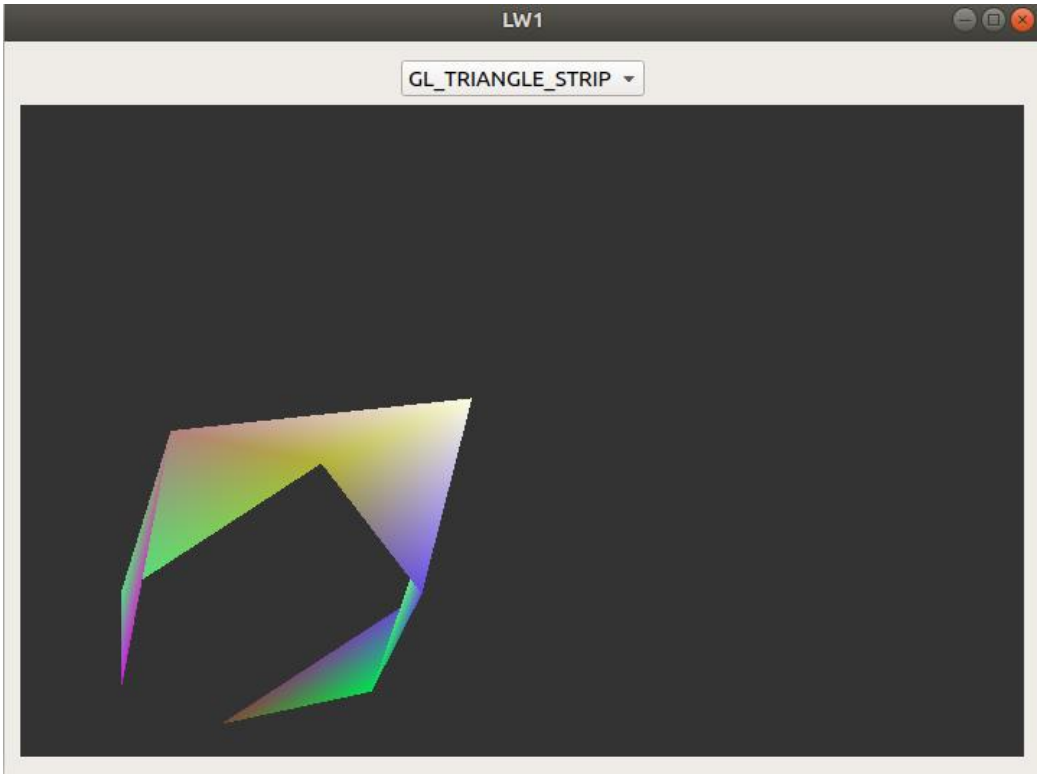


Рисунок 6 – Примитив GL_TRIANGLE_STRIP

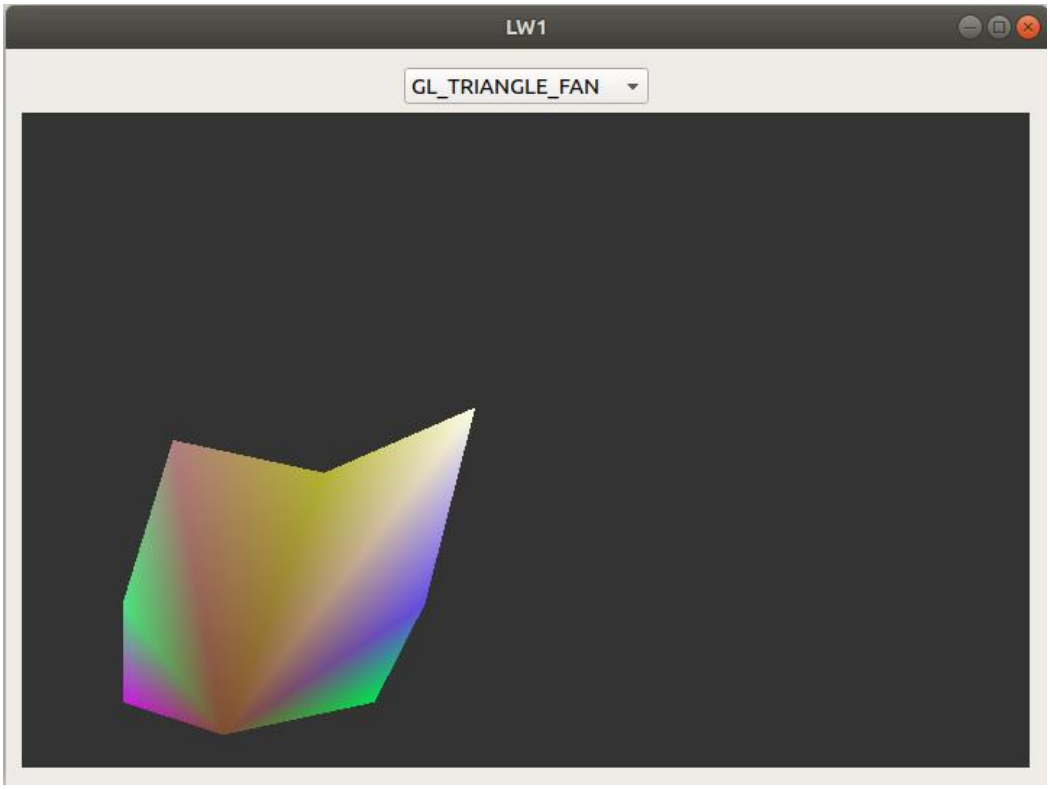


Рисунок 7 – Примитив GL_TRIANGLE_FAN

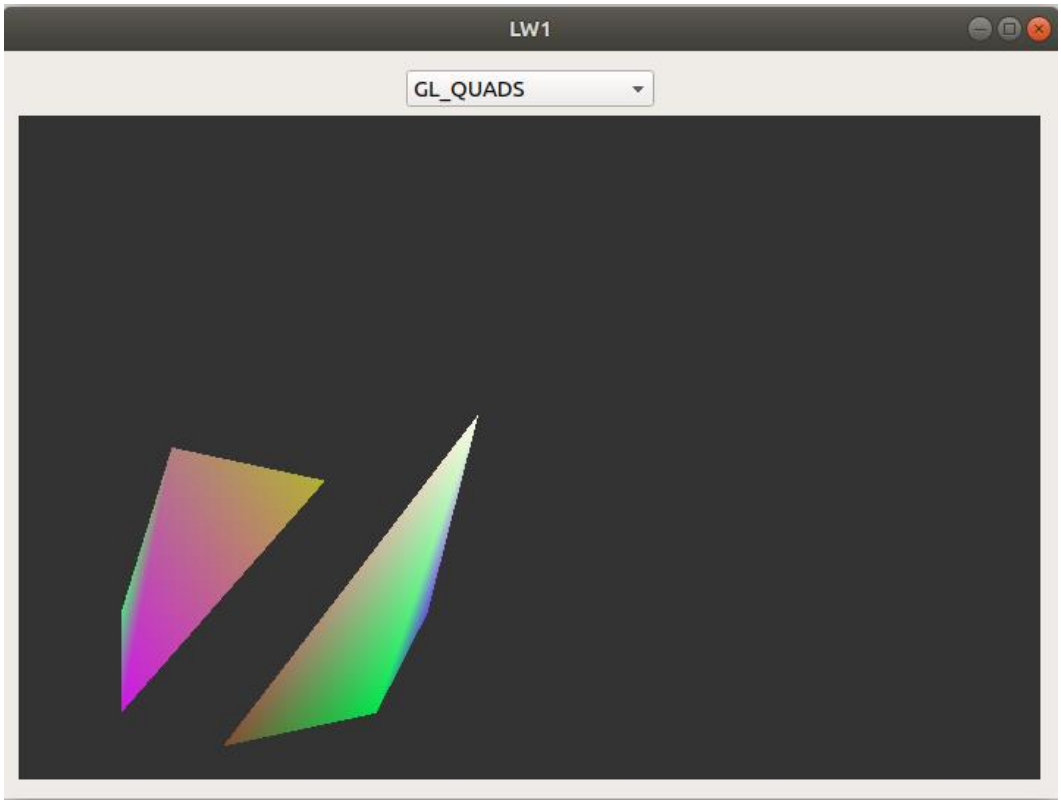


Рисунок 8 – Примитив GL_QUADS

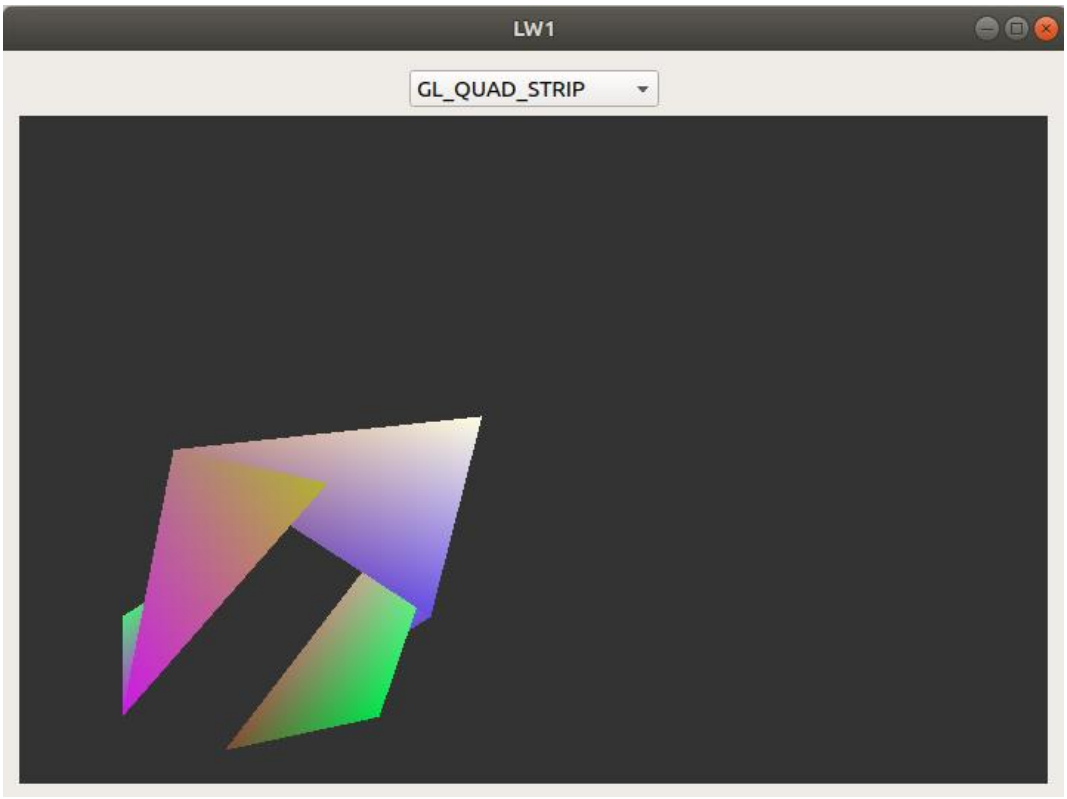


Рисунок 9 – Примитив GL_QUAD_STRIP

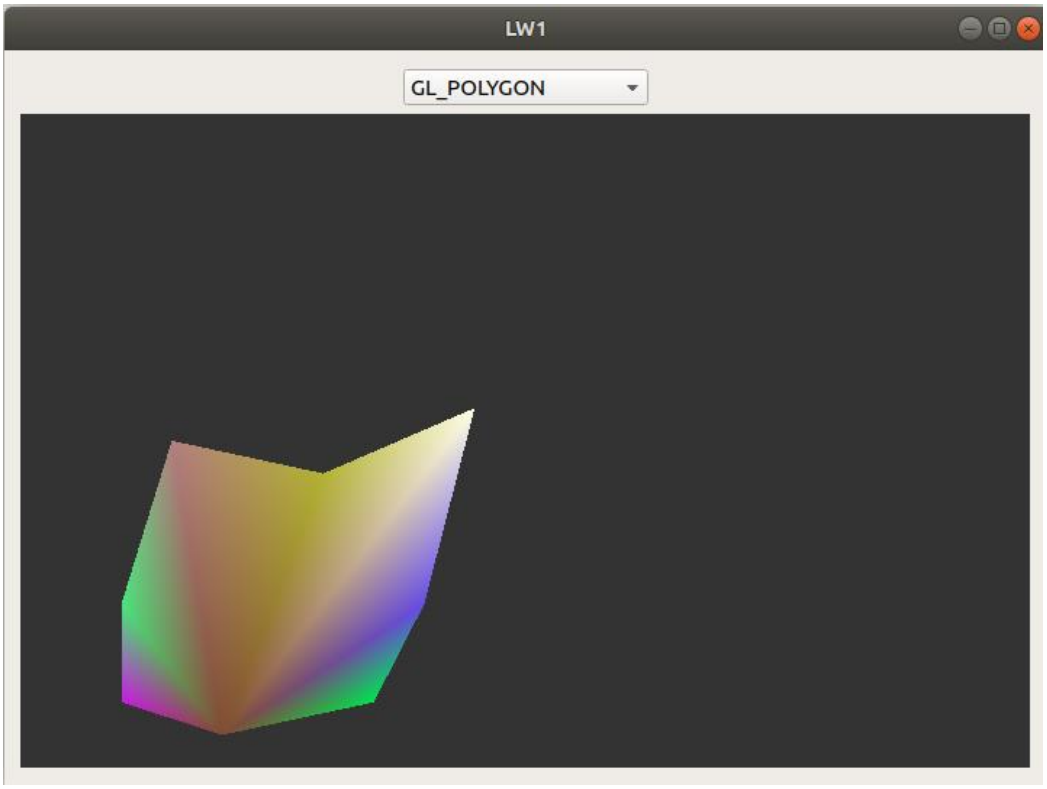


Рисунок 10 – Примитив GL_POLYGON

Выводы.

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.