**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Компьютерная графика»**

Тема:**Примитивы OpenGL**

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

Санкт-Петербург

2020

# Задание

На базе предложенного шаблона разработать программу реализующую представление тестов отсечения ( glScissor), прозрачности (glAlphaFunc), смешения цветов (glBlendFunc) в библиотеке OpenGL на базе разработанных вами в предыдущей работе примитивов.

Разработанная на базе шаблона программа должна быть пополнена возможностями остановки интерактивно различных атрибутов тестов 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.

Рассмотрим метод paintGL.

1. Активируем тесты

## glEnable(GL\_SCISSOR\_TEST);

## glEnable(GL\_ALPHA\_TEST);

## glEnable(GL\_BLEND);

2. Настраиваем viewport и scissor. Viewport здесь используется только для того, чтобы изображение сохраняло соотношение сторон 1:1 при любом соотношении сторон OpenGLWidget и располагалось по центру экрана.

## setupViewport();

## glScissor(

## static\_cast<GLint>(static\_cast<float>(\_w) \* \_scissorX / 2),

## static\_cast<GLint>(static\_cast<float>(\_h) \* \_scissorY / 2),

## static\_cast<GLsizei>(static\_cast<float>(\_w) \* (1 - \_scissorX)),

## static\_cast<GLsizei>(static\_cast<float>(\_h) \* (1 - \_scissorY)));

Функция setupViewport:

## int size = std::min(\_w, \_h);

## glViewport((\_w - size) / 2, (\_h - size) / 2, size, size);

3. Задаем режимы смешивания и функцию тестирования цвета.

## glAlphaFunc(\_alphaTestFunc, \_alphaTestRef);

## glBlendFunc(\_sfactor, \_dfactor);

Дальнейший код мало отличается от представленного в первой лабораторной работе. Добавлены альфа значения цветов, изменены фигуры. Для отрисовки кругов используется функция paintCircleVertices. Для всех кругов используется количество сегментов 360, заданное по умолчанию в заголовочном файле.

## void OpenGLWidget::paintCircleVertices(double cx, double cy, double radius, int segments)

## {

## for(int i = 0; i < segments; i++)

## {

## double theta = 2.0 \* 3.1415926 \* double(i) / double(segments);

## double x = radius \* cos(theta);

## double y = radius \* sin(theta);

## glVertex2d(x + cx, y + cy);

## }

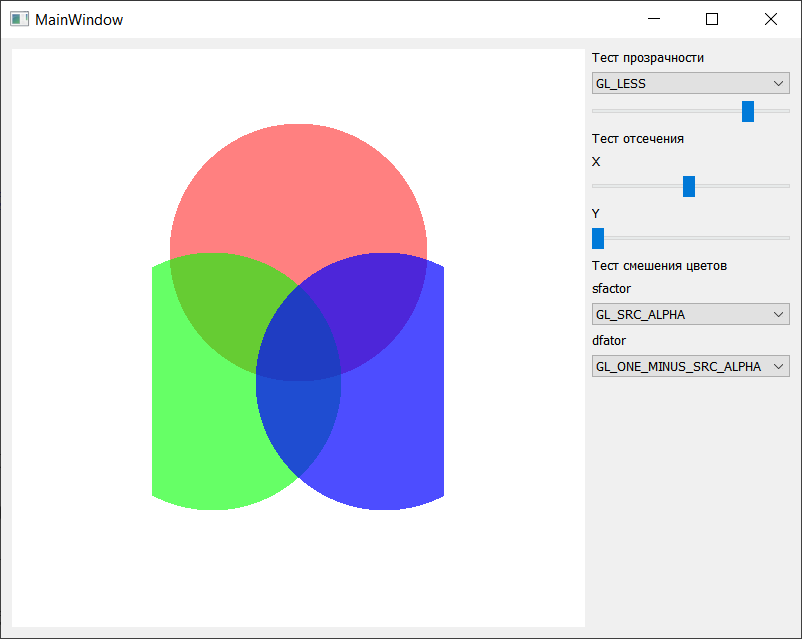
## }

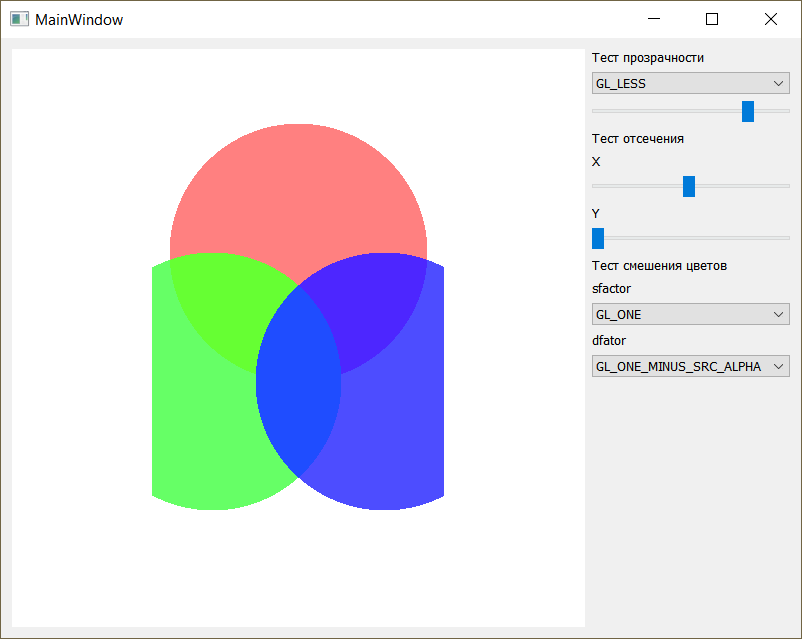
Прозрачности красного, зеленого и синего кругов равны 0.5, 0.6 и 0.7 соответственно.

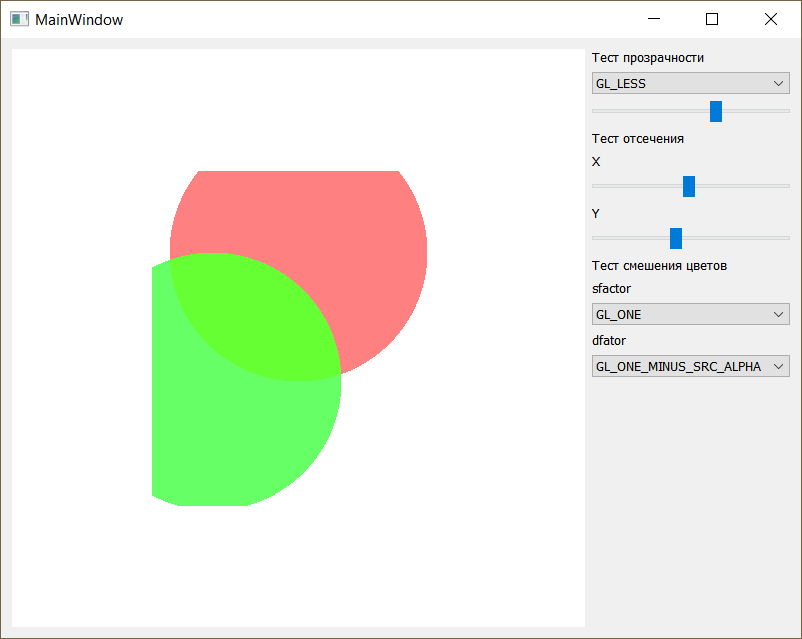
Исходный код: <https://github.com/Pavel-by/sem6_ComputerGraphics/tree/master/lr2/lr2>

# Тестирование

Ссылка на демонстрацию: <https://github.com/Pavel-by/sem6_ComputerGraphics/blob/master/lr2/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.