**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. 8383 |  | Киреев К.А. |
| Преподаватель |  | Герасимова Т.В. |

Санкт-Петербург

2021

**Цель работы.**

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

**Основные теоретические положения.**

Управление режимами работы в OpenGL осуществляется при помощи двух команд - glEnable и glDisable, одна из которых включает, а вторая выключает некоторый режим.

void glEnable(GLenum cap)

void glDisable(GLenum cap)

Обе команды имеют один аргумент – сар, который может принимать значения определяющие тот или иной режим, например, GL\_ALPHA\_TEST, GL\_BLEND, GL\_SCISSOR\_TEST и многие другие.

*Тест отсечения*

Режим GL\_SCISSOR\_TEST разрешает отсечение тех фрагментов объекта, которые находятся вне прямоугольника "вырезки".

Прямоугольник "вырезки" определяется функцией glScissor:

**void glScissor( GLint x, GLint y, GLsizei width, GLsizei height );**

где параметры

* x, y определяют координаты левого нижнего угла прямоугольника «вырезки», исходное значение - (0,0).
* width, height - ширина и высота прямоугольника «вырезки».

В приведенном ниже фрагменте программы реализуется тест отсечения. Сначала изображается группа связных отрезков не используя режим отсечения, а затем включается этот режим.

glEnable(GL\_SCISSOR\_TEST);

InitViewport(0, windH\*2/3, vpW, vpH);

glScissor(0,windH\*2/3,vpW/2,vpH/2);

Triangles();

Quads();

glDisable(GL\_SCISSOR\_TEST);

InitViewport(windW/3, windH\*2/3, vpW, vpH);

glScissor(windW/3,windH\*2/3,vpW/2,vpH/2);

Triangles();

Quads();

*Тест прозрачности*

Режим GL\_ALPHA\_TEST задает тестирование по цветовому параметру альфа.Функция glAlphaFunc устанавливает функцию тестирования параметра альфа.

**void glAlphaFunc( GLenum func, GLclampf ref )**

где параметр – func может принимать следующие значения:

GL\_NEVER – никогда не пропускает

GL\_LESS – пропускает, если входное значение альфа меньше, чем значение ref

GL\_EQUAL – пропускает, если входное значение альфа равно значению ref

GL\_LEQUAL – пропускает, если входное значение альфа меньше или равно значения ref

GL\_GREATER – пропускает, если входное значение альфа больше, чем значение ref

GL\_NOTEQUAL – пропускает, если входное значение альфа не равно значению ref

GL\_GEQUAL – пропускает, если входное значение альфа больше или равно значения ref

GL\_ALWAYS – всегда пропускается, по умолчанию,

а параметр ref – определяет значение, с которым сравнивается входное значение альфа. Он может принимать значение от 0 до 1, причем 0 представляет наименьшее возможное значение альфа, а 1 – наибольшее. По умолчанию ref равен 0.

# В приведенном ниже фрагменте программы реализуется тест прозрачности

glEnable(GL\_ALPHA\_TEST);

InitViewport(windW\*2/3, windH\*2/3, vpW, vpH);

glAlphaFunc(GL\_LESS, 0.7f);

Triangles();

Quads();

InitViewport(0, windH/3, vpW, vpH);

glAlphaFunc(GL\_GREATER, 0.7f);

Triangles();

Quads();

glDisable(GL\_ALPHA\_TEST);

*Тест смешения цветов*

Режим GL\_BLEND разрешает смешивание поступающих значений цветов RGBA со значениями, находящимися в буфере цветов.

Функция glBlendFunc устанавливает пиксельную арифметику.

**void glBlendFunc( GLenum sfactor, GLenum dfactor );**

где параметры

* sfactor устанавливает способ вычисления входящих факторов смешения RGBA. Может принимать одно из следующих значений – GL\_ZERO, GL\_ONE, GL\_DST\_COLOR, GL\_ONE\_MINUS\_DST\_COLOR, GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA, GL\_DST\_ALPHA, GL\_ONE\_MINUS\_DST\_ALPHA и GL\_SRC\_ALPHA\_SATURATE.
* dfactor устанавливает способ вычисления факторов смешения RGBA, уже находящихся в буфере кадра. Может принимать одно из следующих значений – GL\_ZERO, GL\_ONE, GL\_SRC\_COLOR, GL\_ONE\_MINUS\_SRC\_COLOR, GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA, GL\_DST\_ALPHA и GL\_ONE\_MINUS\_DST\_ALPHA.

# В приведенном ниже фрагменте программы реализуется тест смешения

glEnable(GL\_BLEND);

InitViewport(windW/3, windH/3, vpW, vpH);

glBlendFunc(GL\_ONE, GL\_ZERO);

Triangles();

Quads();

InitViewport(windW\*2/3, windH/3, vpW, vpH);

glBlendFunc(GL\_ONE, GL\_ONE);

Triangles();

Quads();

InitViewport(0, 0, vpW, vpH);

glBlendFunc(GL\_ONE, GL\_SRC\_COLOR);

Triangles();

Quads();

InitViewport(windW/3, 0, vpW, vpH);

glBlendFunc(GL\_ONE, GL\_ONE\_MINUS\_SRC\_COLOR);

Triangles();

Quads();

InitViewport(windW\*2/3, 0, vpW, vpH);

glBlendFunc(GL\_ZERO, GL\_ONE\_MINUS\_SRC\_COLOR);

Triangles();

Quads();

Прозрачность лучше организовывать используя команду glBlendFunc(GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA). Такой же вызов применяют для устранения ступенчатости линий и точек. Для устранения ступенчатости многоугольников применяют вызов команды glBlendFunc(GL\_SRC\_ALPHA\_SATURATE, GL\_ONE).

**Выполнение работы.**

Работа была выполнена в среде разработке Qt Creator.

**Подключение библиотеки OpenGL в Qt Creator.**

Для подключения библиотеки OpenGL в среде разработки Qt Creator необходимо в файле Lab2.pro прописать следующий код:

QT += core gui opengl

LIBS += -lglu32 -lopengl32

**Реализация примитивов.**

Для отображения примитивов был переопределен метод класса GLWidget **paintGL()**:

Листинг программы:

void GLWidget::***paintGL***()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glEnable(GL\_SCISSOR\_TEST);

glEnable(GL\_ALPHA\_TEST);

glEnable(GL\_BLEND);

glBlendFunc(*this*->SFactor, *this*->DFactor);

glAlphaFunc(GL\_NEVER+*this*->Func, *this*->Ref/100.0f);

glScissor(*this*->X \* *this*->winSizeW/100,

*this*->Y\**this*->winSizeH /100,

*this*->winSizeW\**this*->Width/100,

*this*->winSizeH\**this*->Height/100) ;

drawPicture(*this*->ComboBox\_type);

glDisable(GL\_BLEND);

glDisable(GL\_ALPHA\_TEST);

glDisable(GL\_SCISSOR\_TEST);

}

В данном методе включаются режимы GL\_SCISSOR\_TEST, GL\_ALPHA\_TEST, GL\_BLEND.

Интерфейс программы представлен на рис. 1.

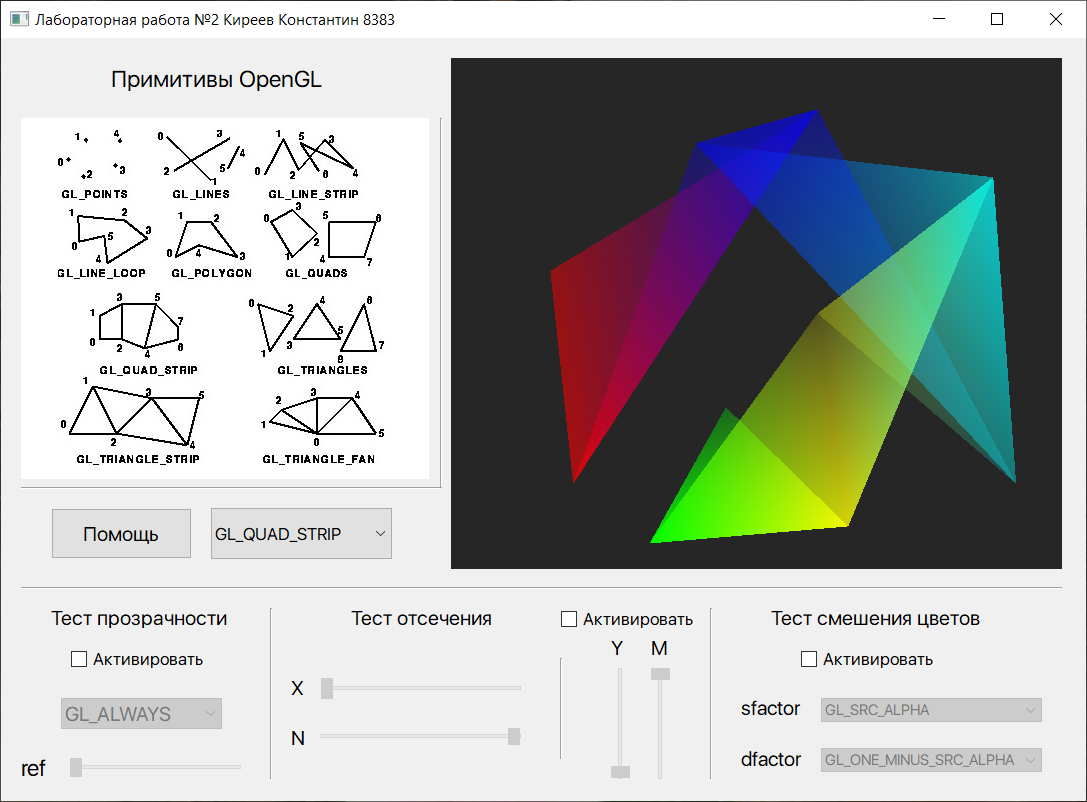


Рис. 1 – Интерфейс программы

Снизу присутствует область, в которой можно настроить отображение. По кнопке помощь можно узнать информацию про примитивы и тесты. Также по флагу активировать выбранный тест можно использовать. В области слева можно настроить параметр для теста прозрачности и выбрать значение альфа. Снизу в центре можно настроить границы отсечения по горизонтали и вертикали. Справа можно настроить значения sfactor и dfactor. Пример работы программы представлен на рис. 2-4.

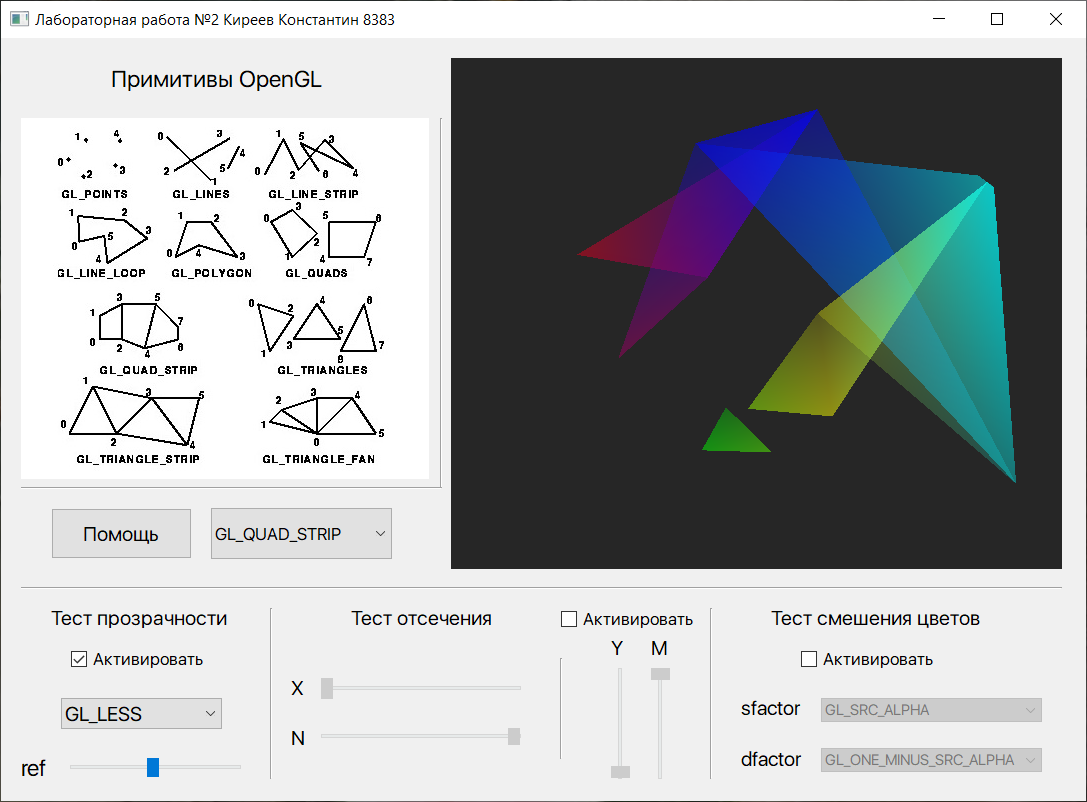


Рис. 2 – Тест прозрачности

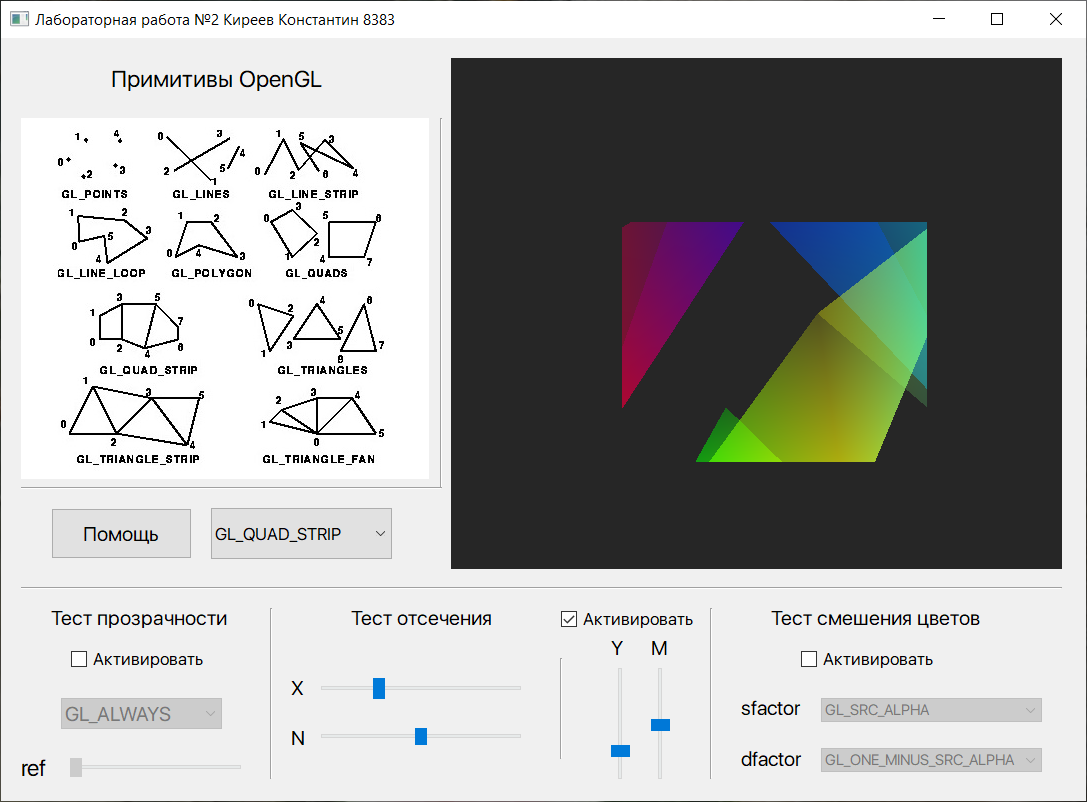


Рис. 3 – Тест отсечения

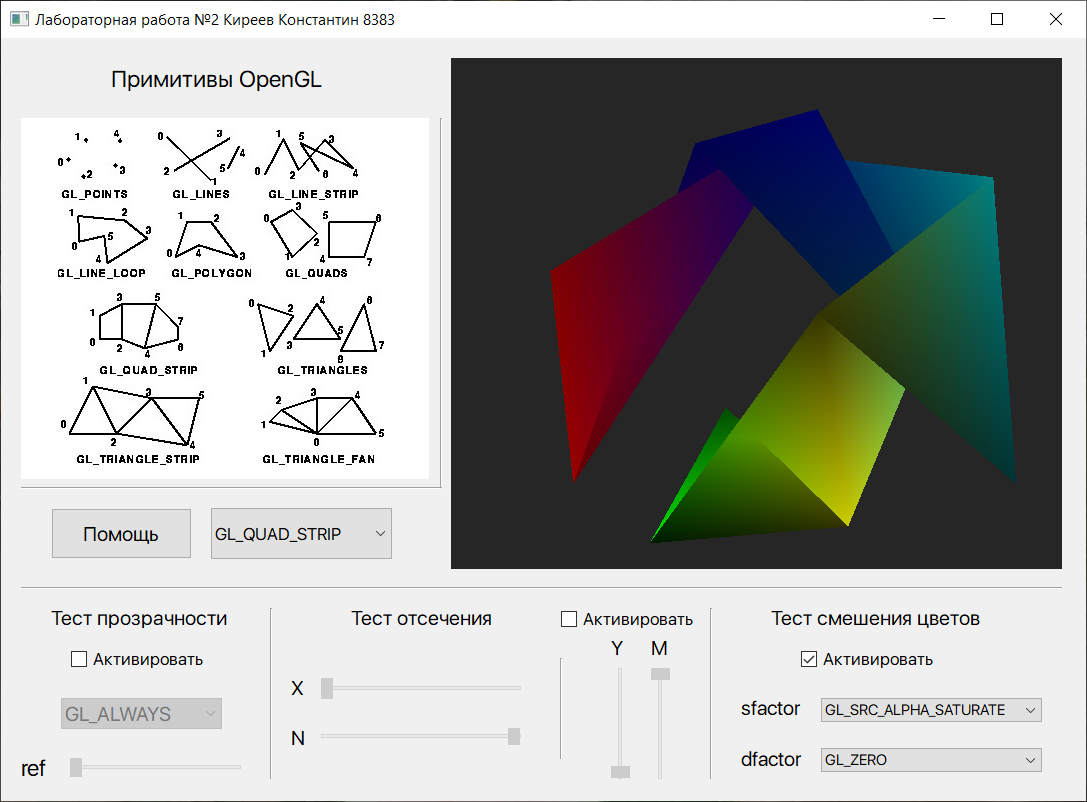


Рис. 4 – Тест смешения цветов

**Выводы.**

В результате выполнения лабораторной работы была разработана программа, реализующая представление определённого набора примитивов из имеющихся в библиотеке OpenGL. Также в программе, при помощи функций glScissor, glAlphaFunc и glBlendFunc, можно применять тесты отсечения, прозрачности и смешения цветов. Программа работает корректно. При выполнении работы были приобретены навыки работы с графической библиотекой OpenGL.