

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. 7383

Власов Р.А.

Преподаватель

Герасимова Т.В.

Санкт-Петербург

2020

Задание.

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

Разработанная программа должна быть пополнена возможностями остановки интерактивно различных атрибутов тестов через вызов соответствующих элементов интерфейса пользователя.

Общие сведения.

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 – cap, 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».

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.

Тест смещения 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 параметр sfactor устанавливает способ вычисления входящих факторов смещения 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.

а `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`.

Ход работы.

Программа разработана на языке программирования Python. Графический интерфейс реализован с помощью библиотеки Qt.

Для успешного отображения примитивов реализован класс `OpenGLView`. В качестве родительского класса взят `QGLWidget`. Перепределены методы `initializeGL`, `resizeGL` и `paintGL`, отвечающие за инициализацию, изменение размера виджета и рисование изображения.

Процесс рисования изображения организован

Активация теста отсечения (`GL_SCISSOR_TEST`) реализована при помощи вызова перед началом рисования следующего кода:

```
glEnable(GL_SCISSOR_TEST)
glScissor(self.scissor[0], self.scissor[1], self.scissor[2],
self.scissor[3])
glBegin(modes.get(self.mode, GL_POINTS))
и после окончания рисования:
glEnd()
glDisable(GL_SCISSOR_TEST)
```

Активация теста прозрачности (`GL_ALPHA_TEST`) реализована при помощи вызова перед началом рисования следующего кода:

```
alpha = {
    0: GL_NEVER,
    1: GL_LESS,
    2: GL_EQUAL,
```

```

3: GL_LEQUAL,
4: GL_GREATER,
5: GL_NOTEQUAL,
6: GL_GEQUAL,
7: GL_ALWAYS
}
glEnable(GL_ALPHA_TEST)
glAlphaFunc(alpha.get(self.alphaTest, GL_NEVER), self.alphaVal)
glBegin(modes.get(self.mode, GL_POINTS))
и после окончания рисования:
glEnd()
glDisable(GL_ALPHA_TEST)

```

Активация теста смещения цветов (GL_BLEND) реализована при помощи вызова перед началом рисования следующего кода:

```

sfactor = {
    0: GL_ZERO,
    1: GL_ONE,
    2: GL_DST_COLOR,
    3: GL_ONE_MINUS_DST_COLOR,
    4: GL_SRC_ALPHA,
    5: GL_ONE_MINUS_SRC_ALPHA,
    6: GL_DST_ALPHA,
    7: GL_ONE_MINUS_DST_ALPHA,
    8: GL_SRC_ALPHA_SATURATE
}
dfactor = {
    0: GL_ZERO,
    1: GL_ONE,
    2: GL_SRC_COLOR,
    3: GL_ONE_MINUS_SRC_COLOR,
    4: GL_SRC_ALPHA,
    5: GL_ONE_MINUS_SRC_ALPHA,
    6: GL_DST_ALPHA,

```

```

7: GL_ONE_MINUS_DST_ALPHA
}
glEnable(GL_BLEND)
glBlendFunc(sfactor.get(self.sfactor, GL_ONE),
dfactor.get(self.dfactor, GL_ZERO))
glBegin(modes.get(self.mode, GL_POINTS))
и после окончания рисования:
glEnd()
glDisable(GL_BLEND)

```

Тестирование.

Программа протестирована в операционной системе Ubuntu 19.04.

Результаты тестирования представлены на рис. 1-10.

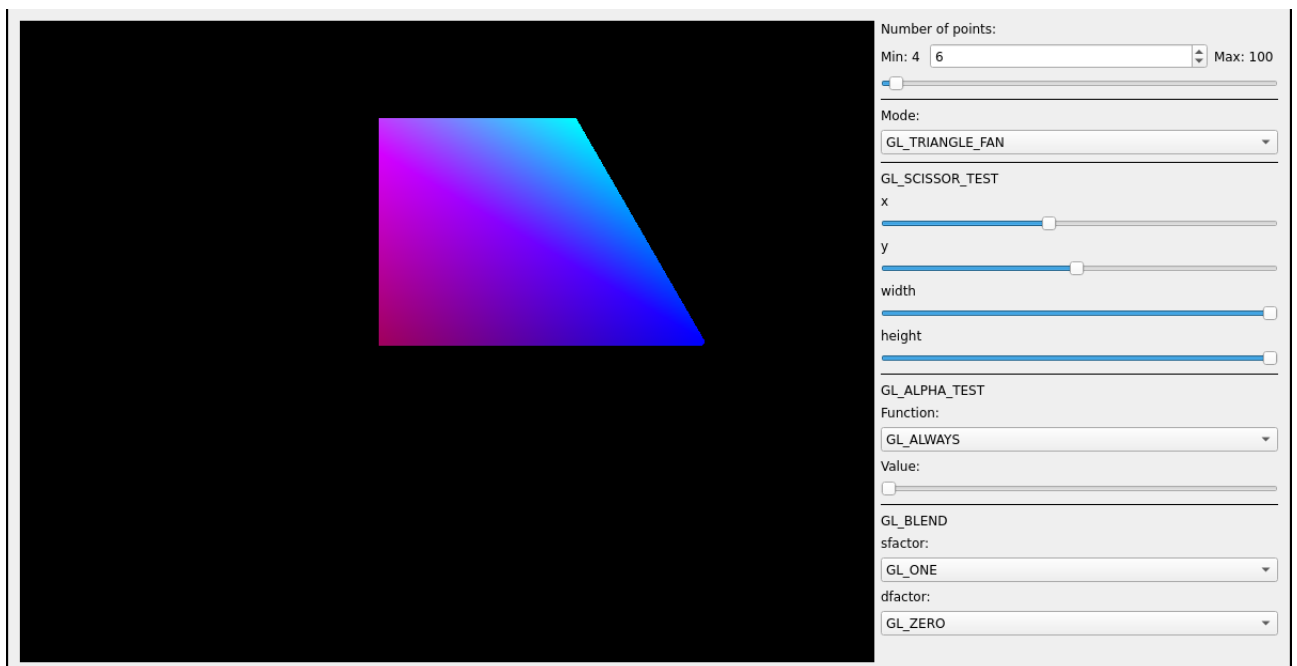


Рисунок 1 – Тестирование GL_SCISSOR_TEST

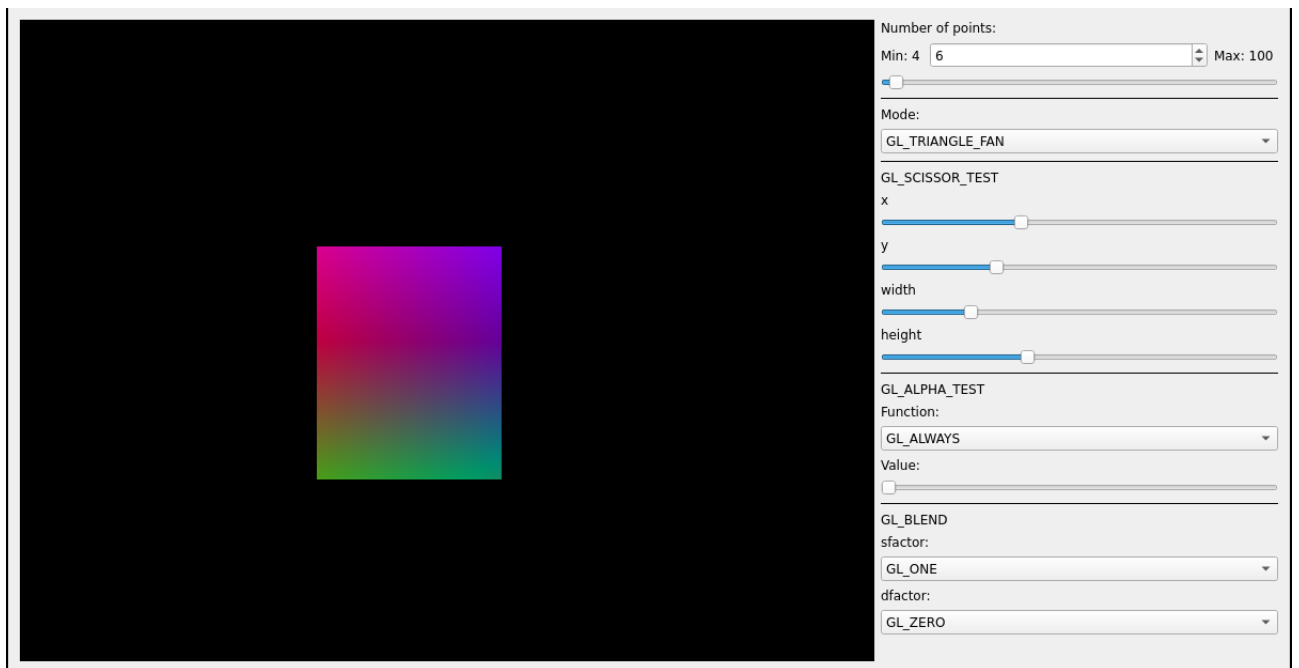


Рисунок 2 – Тестирование GL_SCISSOR_TEST

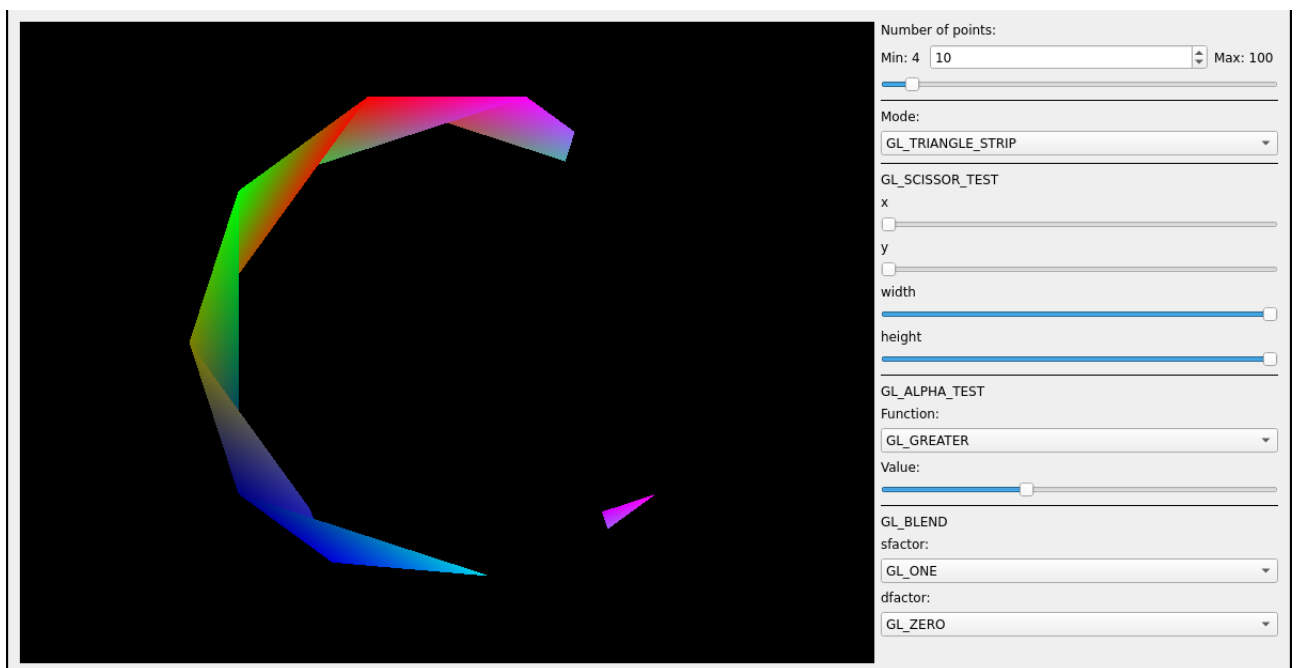


Рисунок 3 – Тестирование GL_ALPHA_TEST

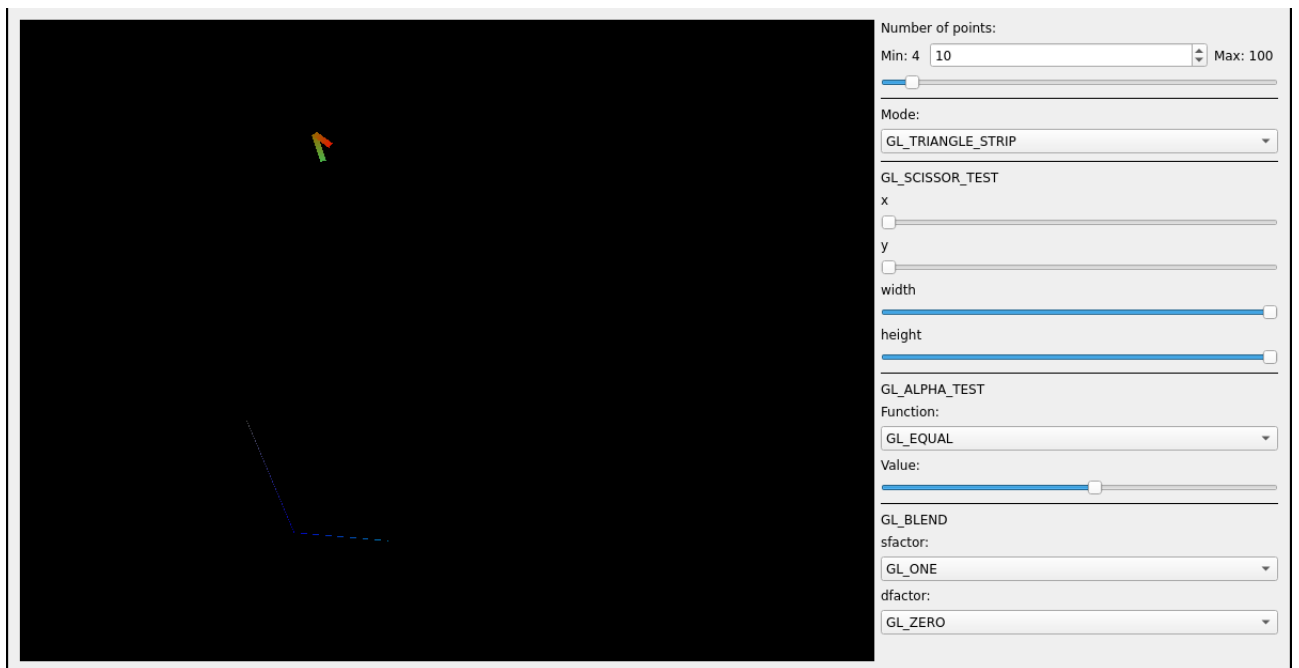


Рисунок 4 – Тестирование GL_ALPHA_TEST

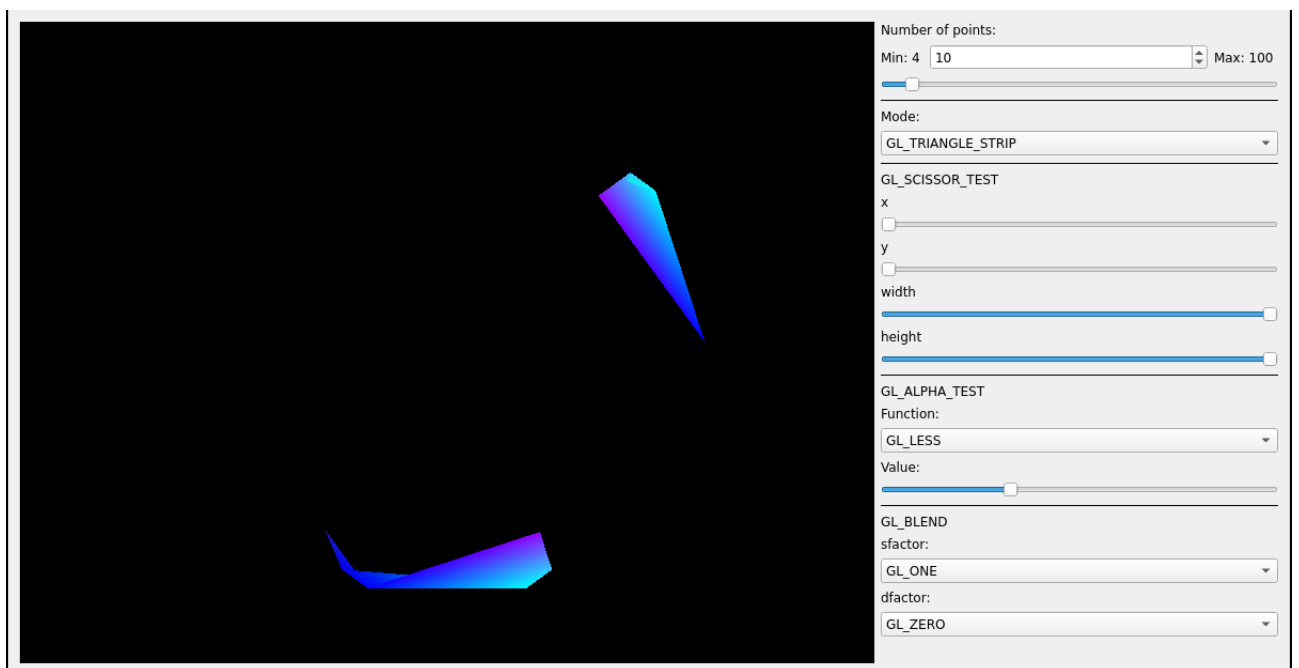


Рисунок 5 – Тестирование GL_ALPHA_TEST

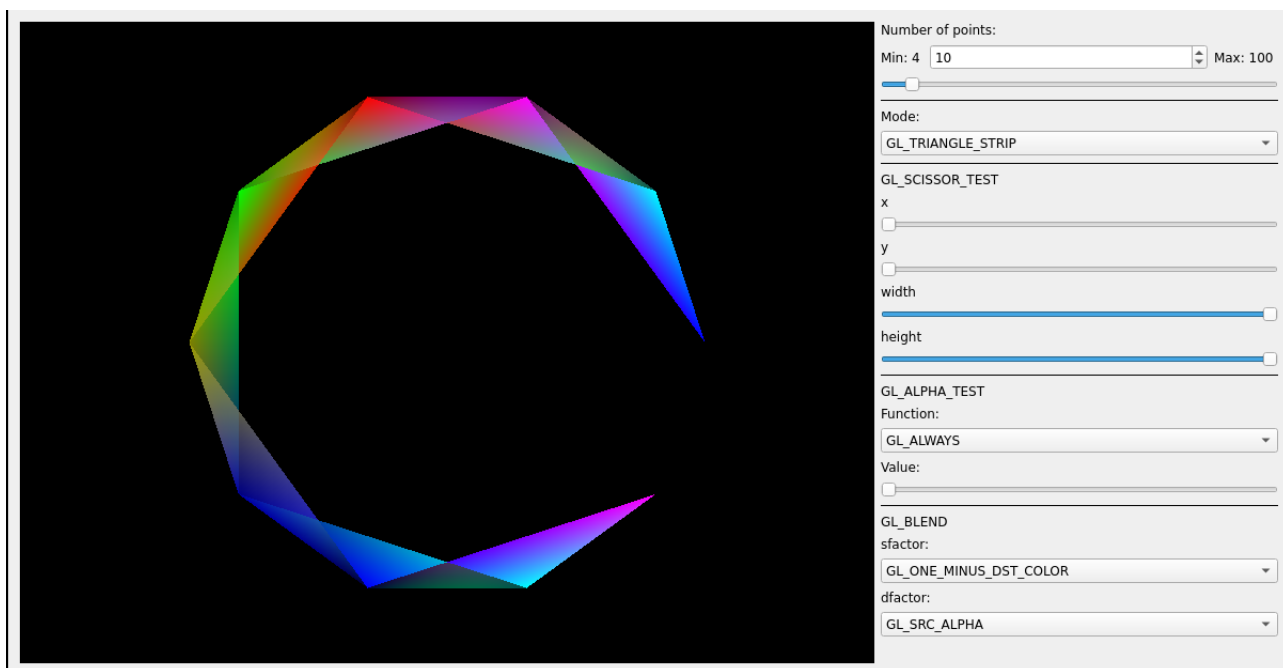


Рисунок 6 – Тестирование GL_BLEND

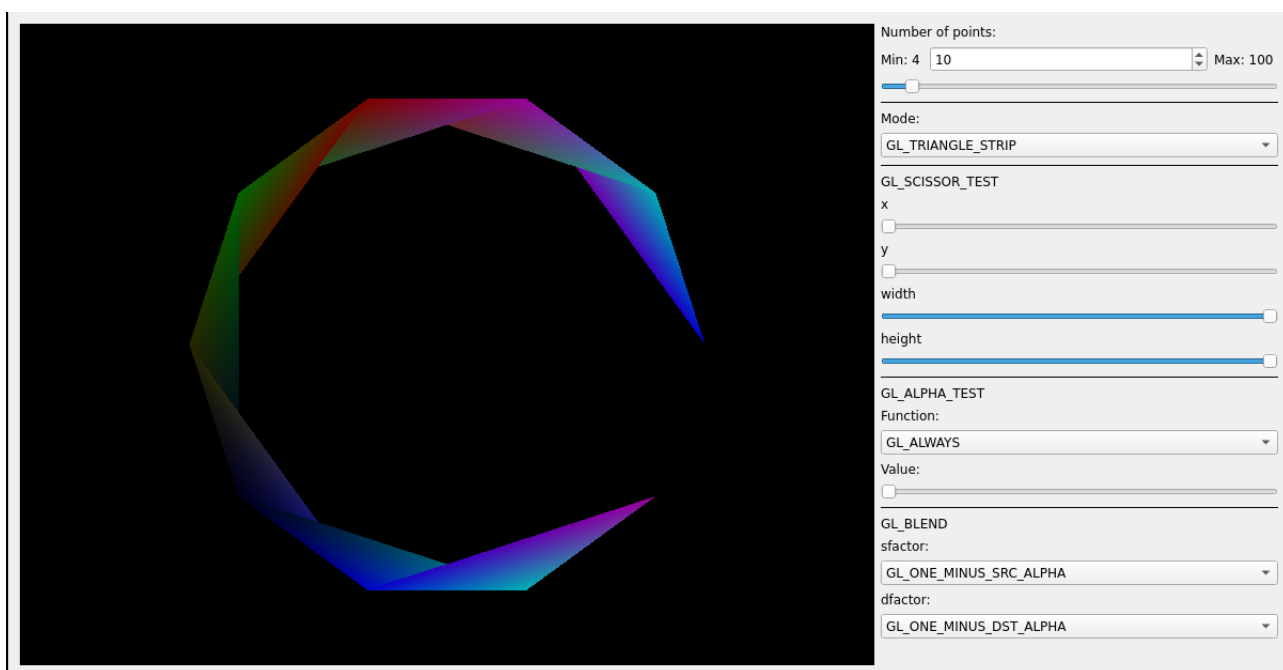


Рисунок 7 – Тестирование GL_BLEND

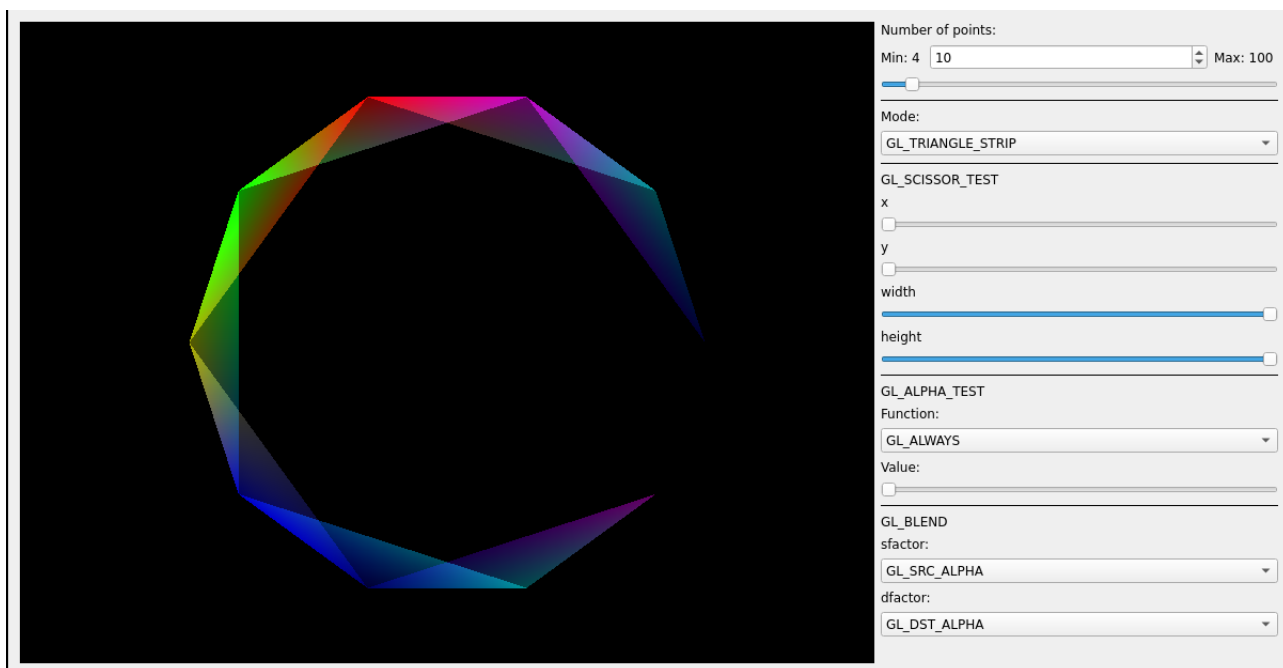


Рисунок 8 – Тестирование GL_BLEND

Вывод.

В результате выполнения лабораторной работы была разработана программа, реализующая представление тестов смешивания цветов, отсечения и прозрачности для графических примитивов OpenGL, разработанных в лабораторной работе №1. Программа работает корректно. При выполнении работы были приобретены навыки работы с графической библиотекой OpenGL.