

**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: Прimitives OpenGL**

Студент гр. 0304

\_\_\_\_\_

Максименко Е.М.

Преподаватель

\_\_\_\_\_

Герасимова Т.В.

Санкт-Петербург

2023

### **Цель работы.**

- ознакомление с основными примитивами OpenGL.
- освоение возможности подключения графической библиотеки в среду разработки.

### **Задание.**

На базе разработанной оболочки из 1 работы разработать программу реализующую представление тестов отсечения ( `glScissor`), прозрачности (`glAlphaFunc`), смешения цветов (`glBlendFunc`) в библиотеке OpenGL на базе разработанных в предыдущей работе примитивов.

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

### **Выполнение работы.**

Работа была выполнена с использованием языка программирования C++ и фреймворка Qt 6. Каркасом программы послужила программа из работы 1.

Класс виджета для работы с OpenGL `GLScene` был модифицирован (см. рис. 1). В класс были добавлены новые методы для изменения параметров теста отсечения, теста прозрачности и теста смешения цветов. Также создание точек для примитивов было вынесено в метод *setPoints*.

```

7  class GLScene: public QOpenGLWidget, public QOpenGLFunctions
8  {
9      Q_OBJECT
10 public:
11     GLScene(QWidget* parent = nullptr);
12
13     void setPrimitiveMode(GLenum mode);
14
15     void setAlphaTestMode(GLenum mode);
16     void setAlphaRef(GLdouble ref);
17
18     void setBlendSFactor(GLenum factor);
19     void setBlendDFactor(GLenum factor);
20
21     void setScissorXRate(GLdouble rate);
22     void setScissorYRate(GLdouble rate);
23     void setScissorWRate(GLdouble rate);
24     void setScissorHRate(GLdouble rate);
25 protected:
26     virtual void initializeGL() override;
27     virtual void resizeGL(int w, int h) override;
28     virtual void paintGL() override;
29 private:
30     void setPoints();
31
32     GLint frameWidth      = 0;
33     GLint frameHeight     = 0;
34
35     GLenum primitiveMode  = GL_POINTS;
36
37     bool alphaTestEnabled  = true;
38     GLenum alphaTest       = GL_ALWAYS;
39     // from 0 to 1
40     GLdouble alpha         = 0.5f;
41
42     bool blendTestEnabled  = true;
43     GLenum sFactorBlend    = GL_ONE;
44     GLenum dFactorBlend    = GL_ZERO;
45
46     bool scissorTestEnabled = true;
47     // from 0 to 1
48     GLdouble scissorXRate  = 0.f;
49     GLdouble scissorYRate  = 0.f;
50     GLdouble scissorWRate  = 1.f;
51     GLdouble scissorHRate  = 1.f;
52 };

```

Рисунок 1. Устройство класса GLScene

Также был изменен метод *paintGL*. Его код представлен в листинге 1. В метод были добавлены включение и отключение теста отсечения, теста прозрачности и теста смешения цветов.

Листинг 1. Код метода *paintGL*.

```
void GLScene::paintGL()
{
    // enable test modes
    if (scissorTestEnabled)
        glEnable(GL_SCISSOR_TEST);
    if (alphaTestEnabled)
        glEnable(GL_ALPHA_TEST);
    if (blendTestEnabled)
        glEnable(GL_BLEND);

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glPointSize(5.f);
    glLineWidth(3.f);

    glScissor(
        scissorXRate * frameWidth,
        scissorYRate * frameHeight,
        scissorWRate * (1.f - scissorXRate) * frameWidth,
        scissorHRate * (1.f - scissorYRate) * frameHeight
    );
    glAlphaFunc(alphaTest, alpha);
    glBlendFunc(sFactorBlend, dFactorBlend);

    glBegin(primitiveMode);
        setPoints();
    glEnd();

    // disable test modes
    if (scissorTestEnabled)
        glDisable(GL_SCISSOR_TEST);
    if (alphaTestEnabled)
        glDisable(GL_ALPHA_TEST);
    if (blendTestEnabled)
        glDisable(GL_BLEND);
}
```

Был добавлен вызов функции *glScissor* для теста отсечения. В данную функцию передаются координаты левого нижнего угла прямоугольной области отсечения, а также размеры данной области.

Был добавлен вызов функции *glAlphaFunc* для теста прозрачности. Функция принимает тип тестирования, а также значение, относительно которого будет проводиться тестирование.

Был добавлен вызов функции *glBlendFunc* для теста смешения цветов. Функция принимает два параметра: способ вычисления входящих факторов смешения RGBA и способ вычисления факторов смешения RGBA, находящихся в буфере кадра.

Также был расширен пользовательский интерфейс. Пользователь может настроить все параметры тестирования, и виджет перерисовывается «на лету».

## Тестирование.

Программа была протестирована для различных настроек тестов отсечения, прозрачности и смешения цветов. Результаты тестирования см. на рис. 2 — 10.

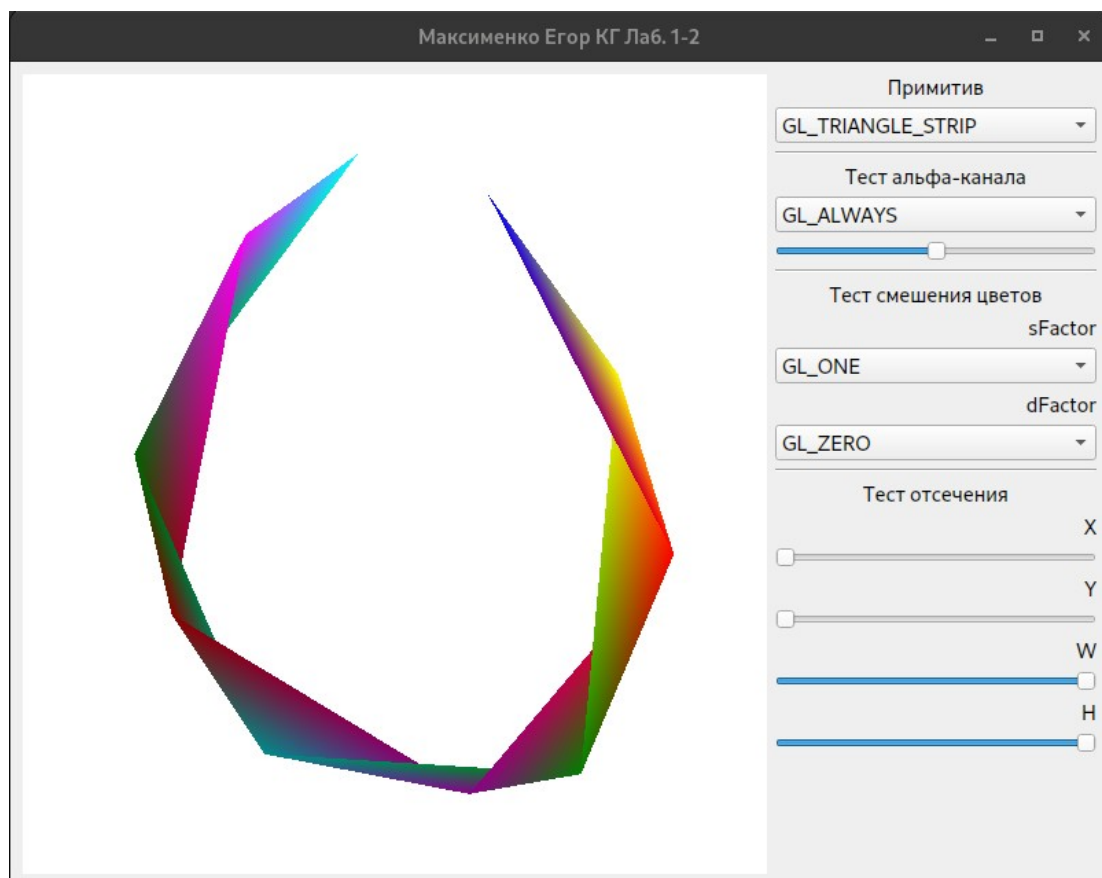


Рисунок 2. Запуск программы при настройках по умолчанию

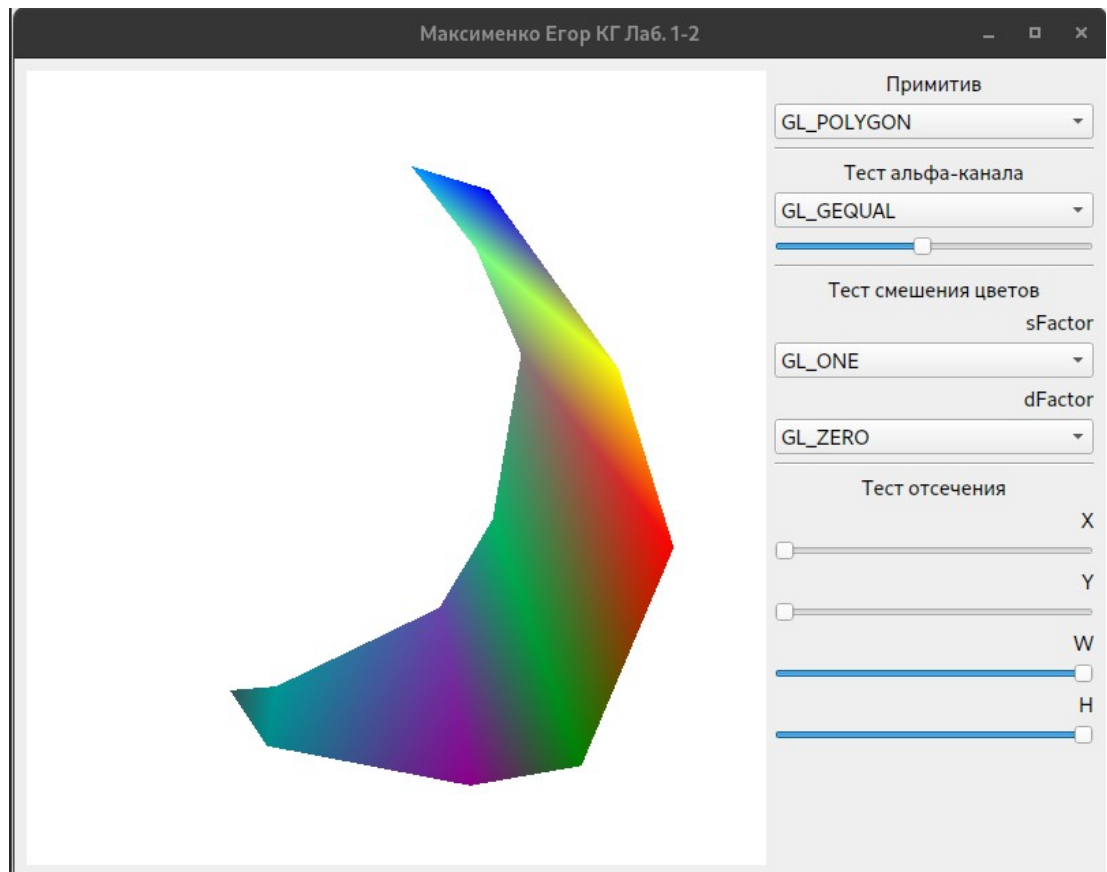


Рисунок 3. Результаты запуска с тестом отсечения GL\_EQUAL

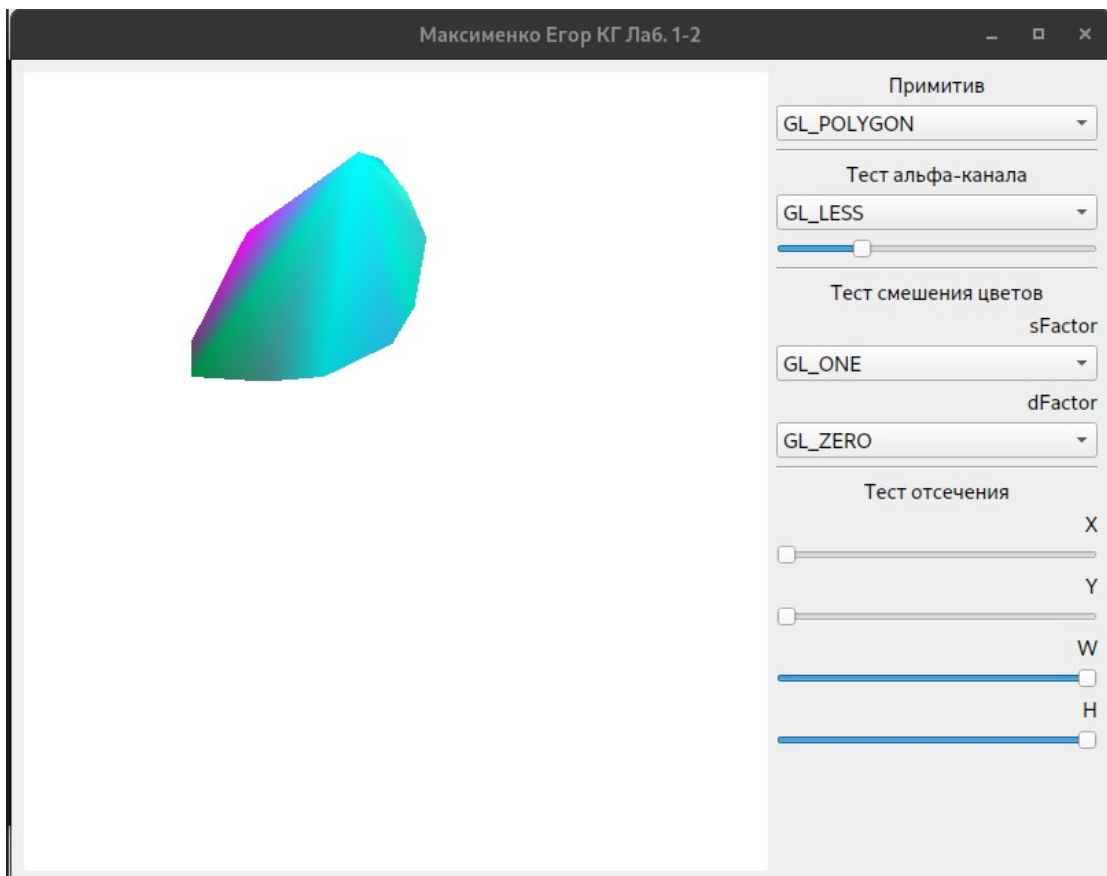


Рисунок 4. Результаты запуска с тестом отсечения GL\_LESS

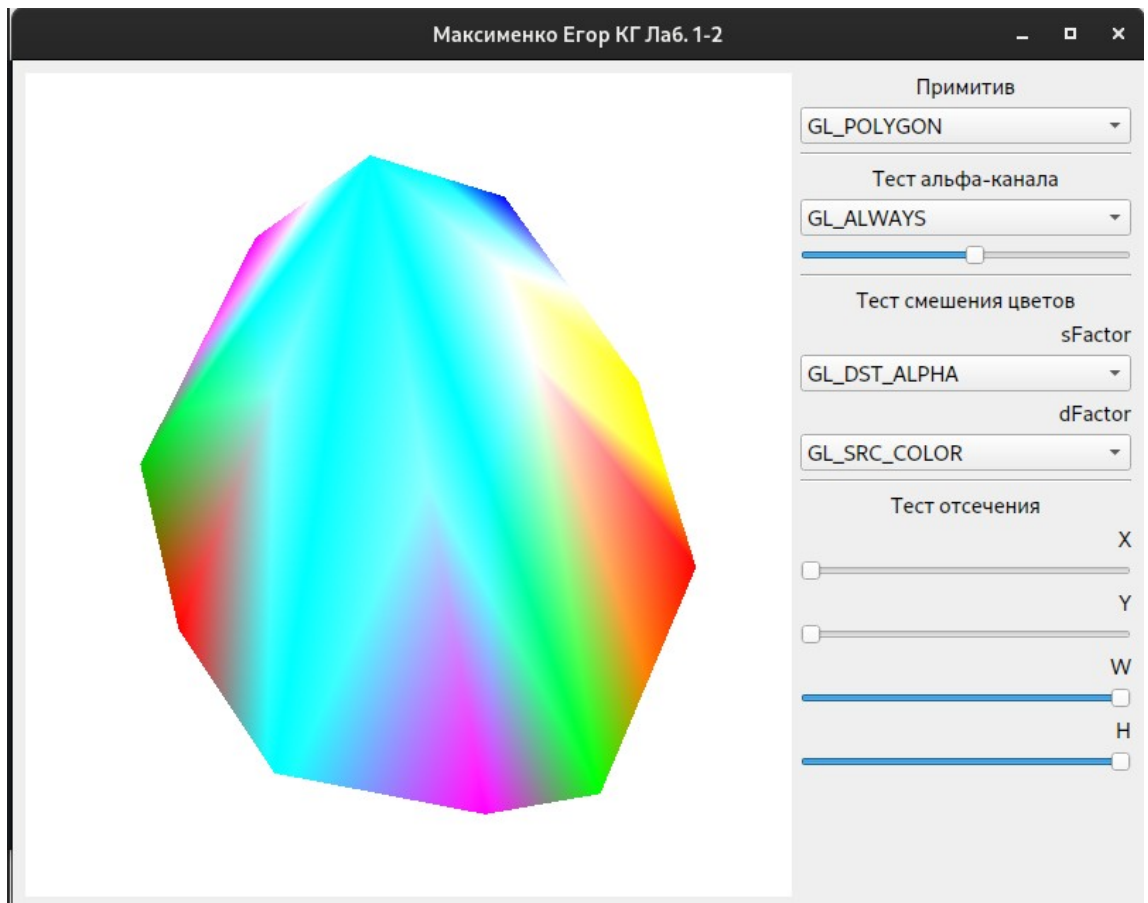


Рисунок 5. Результаты запуска с тестом смешения  $s\text{factor}=\text{GL\_DST\_ALPHA}$ ,  
 $d\text{factor}=\text{GL\_SRC\_COLOR}$

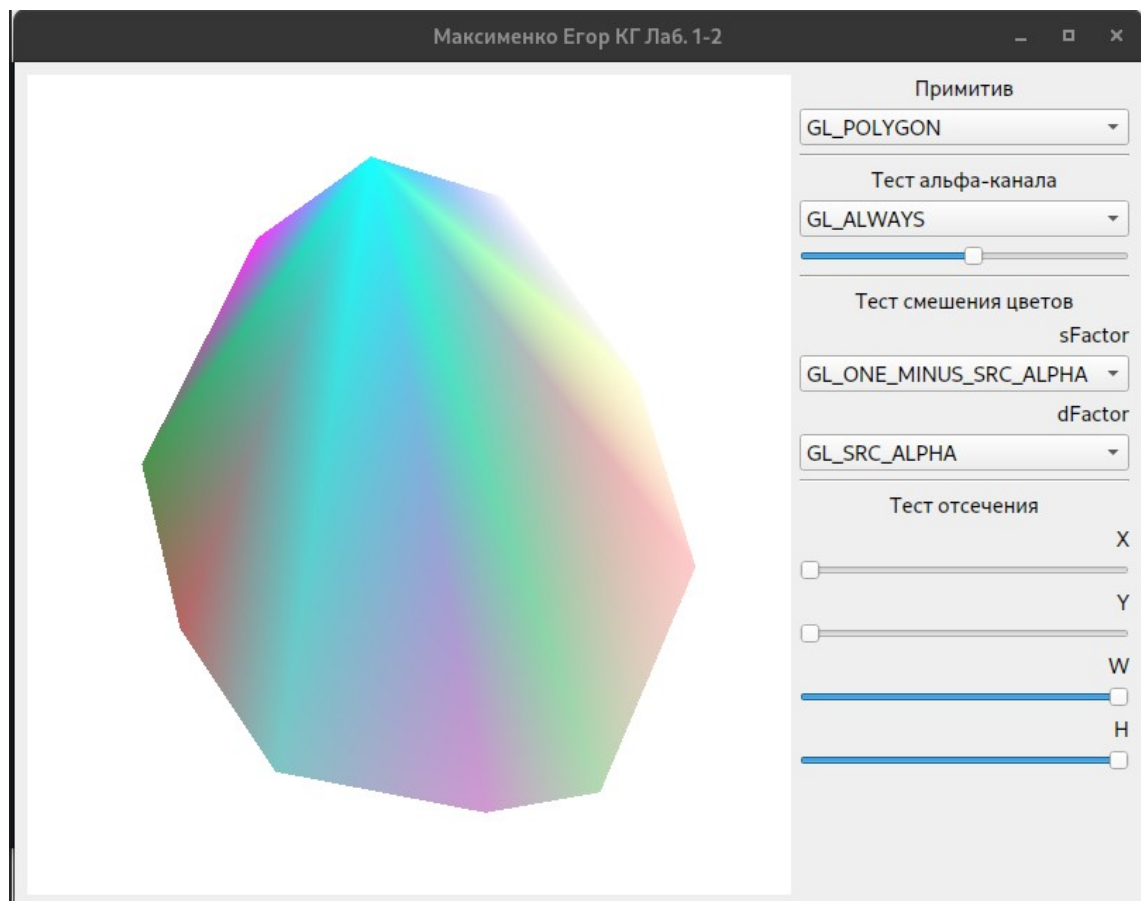


Рисунок 6. Результаты запуска с тестом смешения  $\text{sfactor}=\text{GL\_ONE\_MINUS\_SRC\_ALPHA}$ ,  
 $\text{dfactor}=\text{GL\_SRC\_ALPHA}$



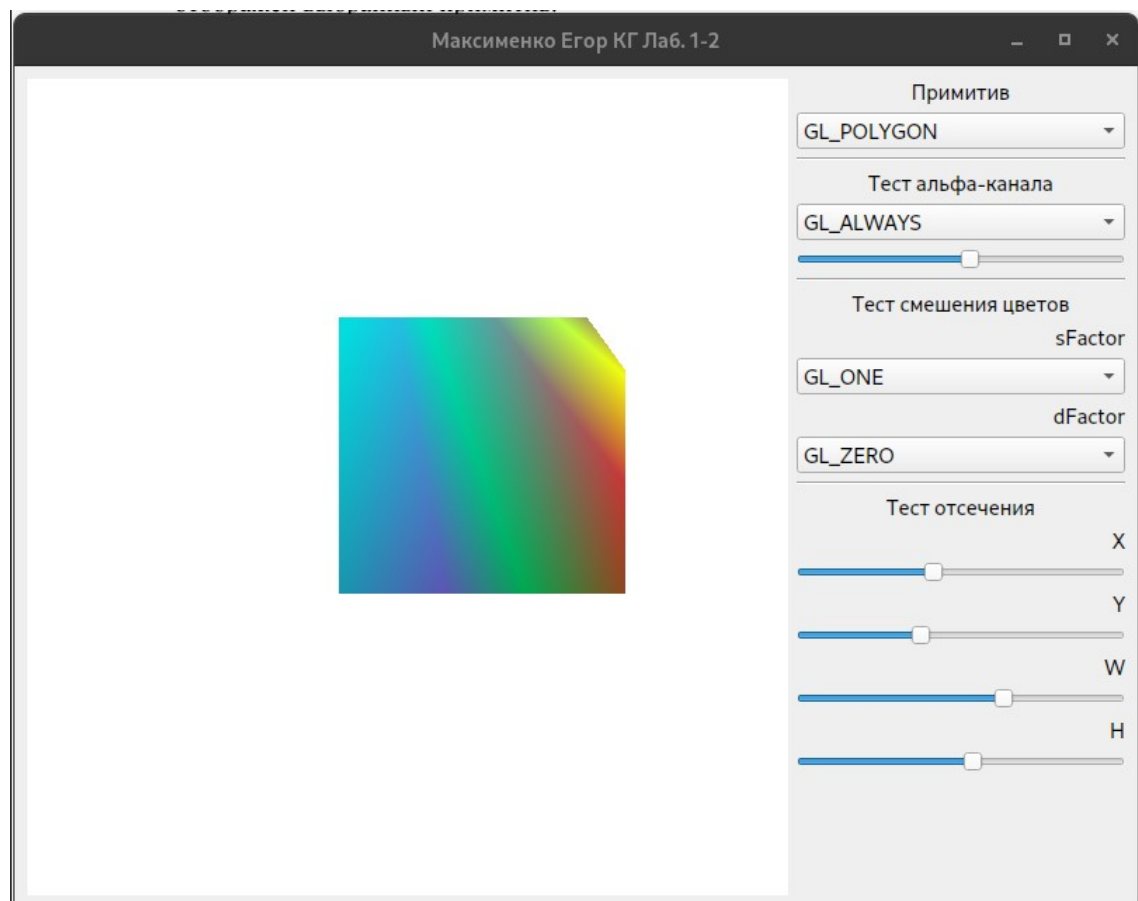


Рисунок 7. Результаты запуска с тестом отсечения

## 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.