

**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. 9304

Попов Д.С.

Преподаватель

Герасимова Т.В.

Санкт-Петербург

2022

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

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».

В приведенном ниже фрагменте программы реализуется тест отсечения. Сначала изображается группа связанных 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 выполнена в среде разработки Qt.

Для выбора параметров каждого теста на боковой панели окна приложения создана кнопка с всплывающим списком. Изначально значения параметров выставлены в `GL_ALWAYS`, `GL_ZERO` и `GL_ONE`. При изменении какого-либо параметра вызывается соответствующий данному тесту метод класса `IFigure`, от которого унаследованы классы, реализующие фигуры.

```

77 void IFigure::setAlphaTest(GLuint test, double value)
78 {
79     alphaTest = test;
80     alphaTestValue = value;
81     nFigure = draw(1.2f);
82     update();
83 }
84
85 void IFigure::setBlenderTest(GLuint sF, GLuint dF)
86 {
87     blenderTestDFactor = dF;
88     blenderTestSFactor = sF;
89     nFigure = draw(1.2f);
90     update();
91 }
92
93 void IFigure::setScissorTest(GLint _x, GLint _y, GLsizei _w, GLsizei _h)
94 {
95     x = _x;
96     y = _y;
97     w = _w;
98     h = _h;
99     nFigure = draw(1.2f);
100    update();
101 }

```

В каждом методе происходит инициализация соответствующих полей класса:

```

protected:
    GLuint alphaTest = GL_ALWAYS;           ///< режим теста альфа-канала.
    double alphaTestValue = 0.0;           ///< значения сравнения для альфа-теста.
    GLint x = 0;                           ///< координаты x угла для теста отсечения.
    GLint y = 0;                           ///< координаты y угла для теста отсечения.
    GLsizei w = width();                   ///< ширина отсекающего прямоугольника.
    GLsizei h = width();                   ///< ширина отсекающего прямоугольника.
    GLuint blenderTestSFactor = GL_ZERO;    ///< способ вычисления входящих факторов смешения RGBA.
    GLuint blenderTestDFactor = GL_ONE;    ///< способ вычисления факторов смешения RGBA, уже находящихся в буфере кадра.

```

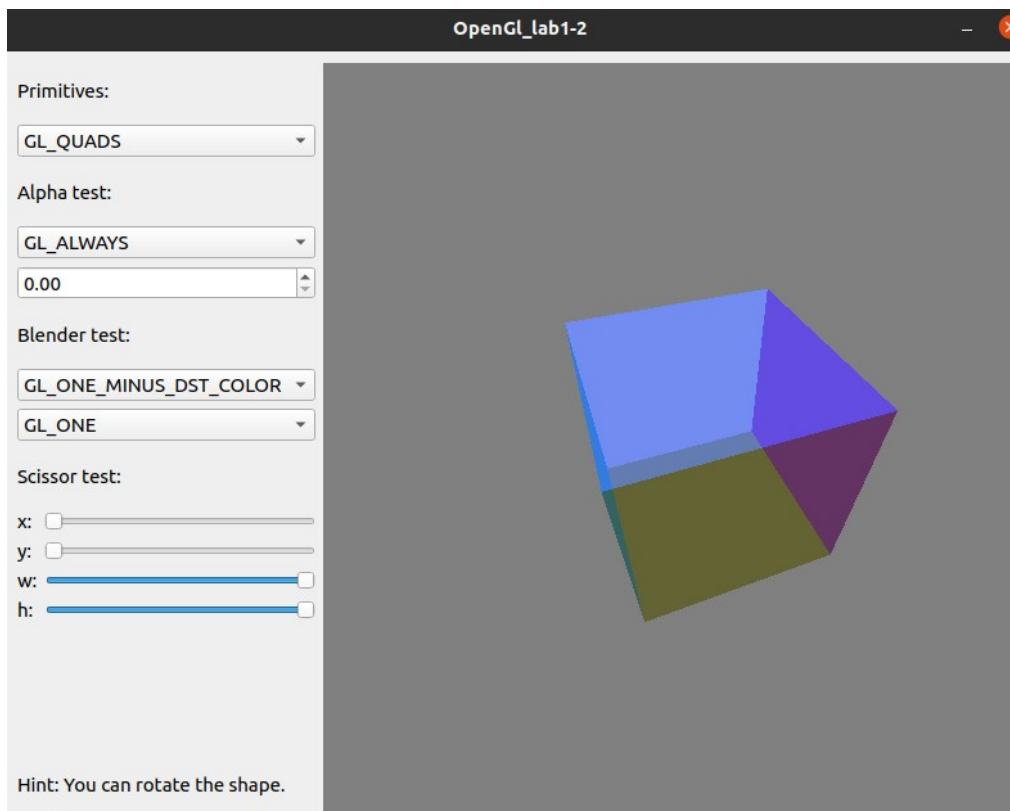
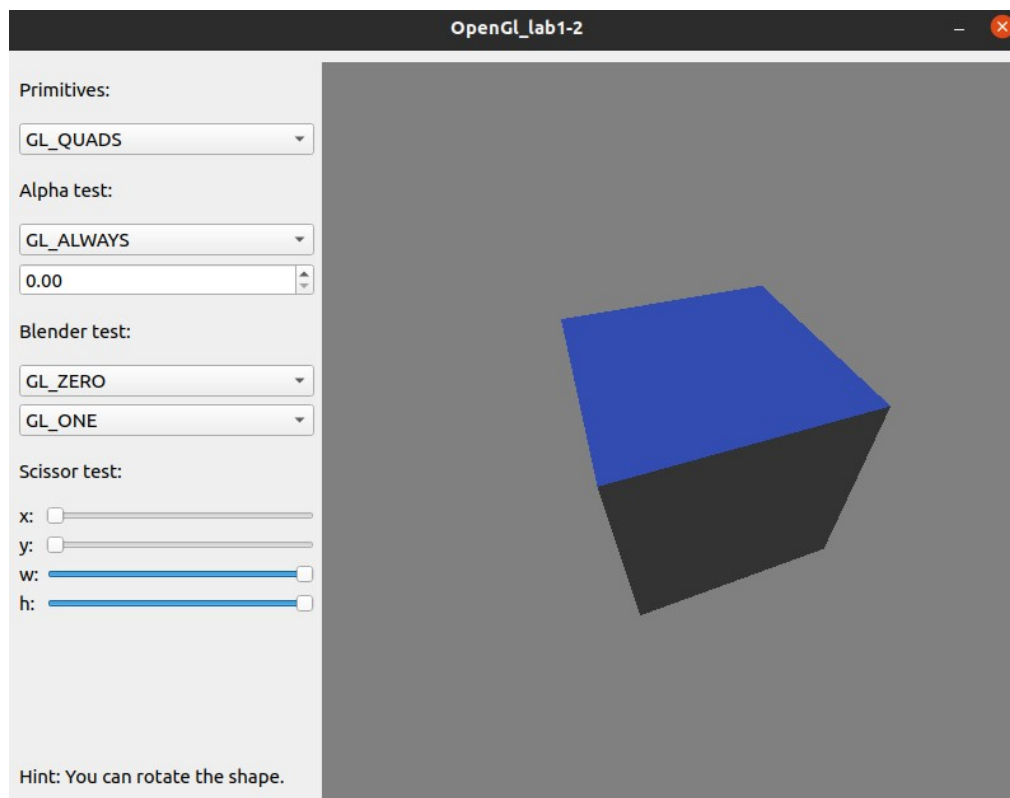
В методе draw класса IFigure происходит включение и выключение режимов GL_SCISSOR_TEST, GL_ALPHA_TEST и GL_BLEND, а также вызов функций glScissor, glAlphaFunc и glBlendFunc.

```
21  GLuint IFigure::draw(GLfloat size)
22  {
23      GLuint n = glGenLists(1);
24      glNewList(n, GL_COMPILE);
25
26      glEnable(GL_BLEND);
27      glEnable(GL_ALPHA_TEST);
28      glEnable(GL_SCISSOR_TEST);
29
30      glBlendFunc(blenderTestDFactor, blenderTestSFactor);
31      glAlphaFunc(alphaTest, alphaTestValue);
32      glScissor(x, y, w, h);
33
34      createFigure(size);
35
36      glDisable(GL_BLEND);
37      glDisable(GL_ALPHA_TEST);
38      glDisable(GL_SCISSOR_TEST);
39
40      glEndList();
41
42      return n;
43  }
```

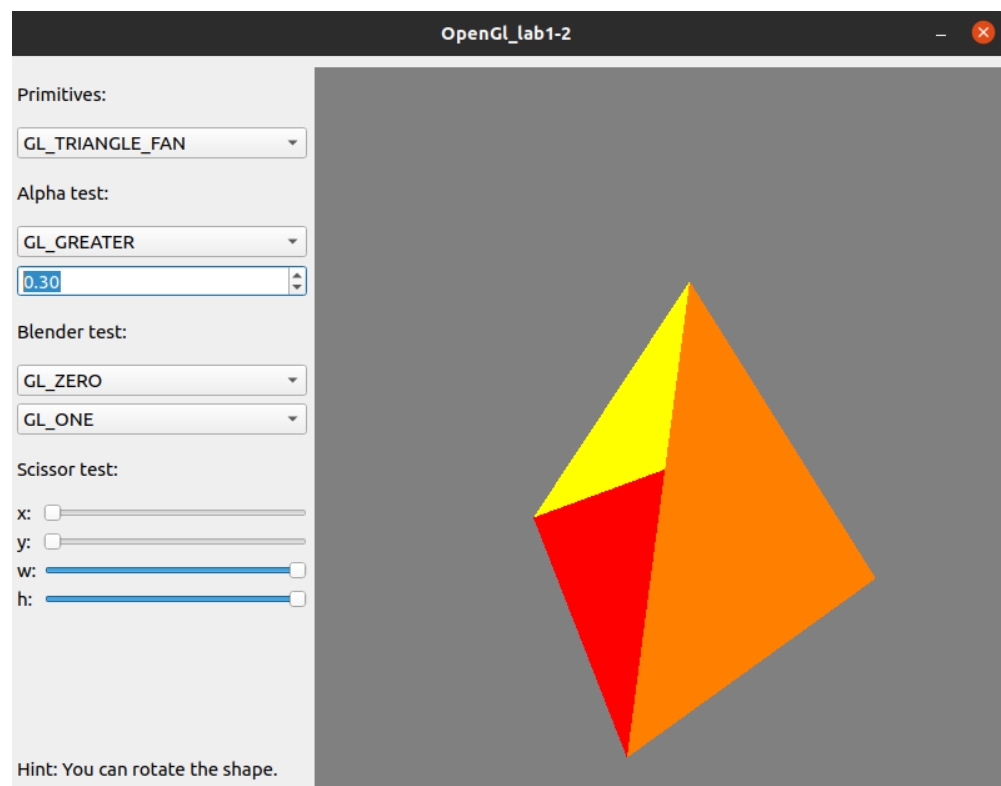
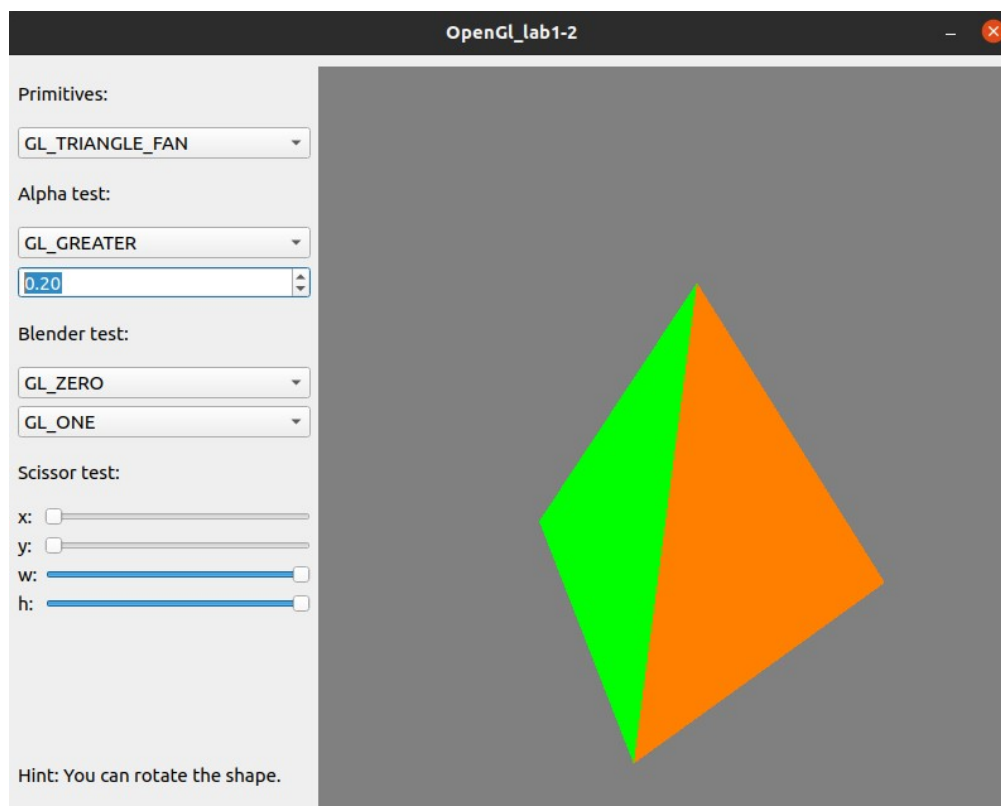
Тестирование

Результаты тестирования представлены на снимках экрана.

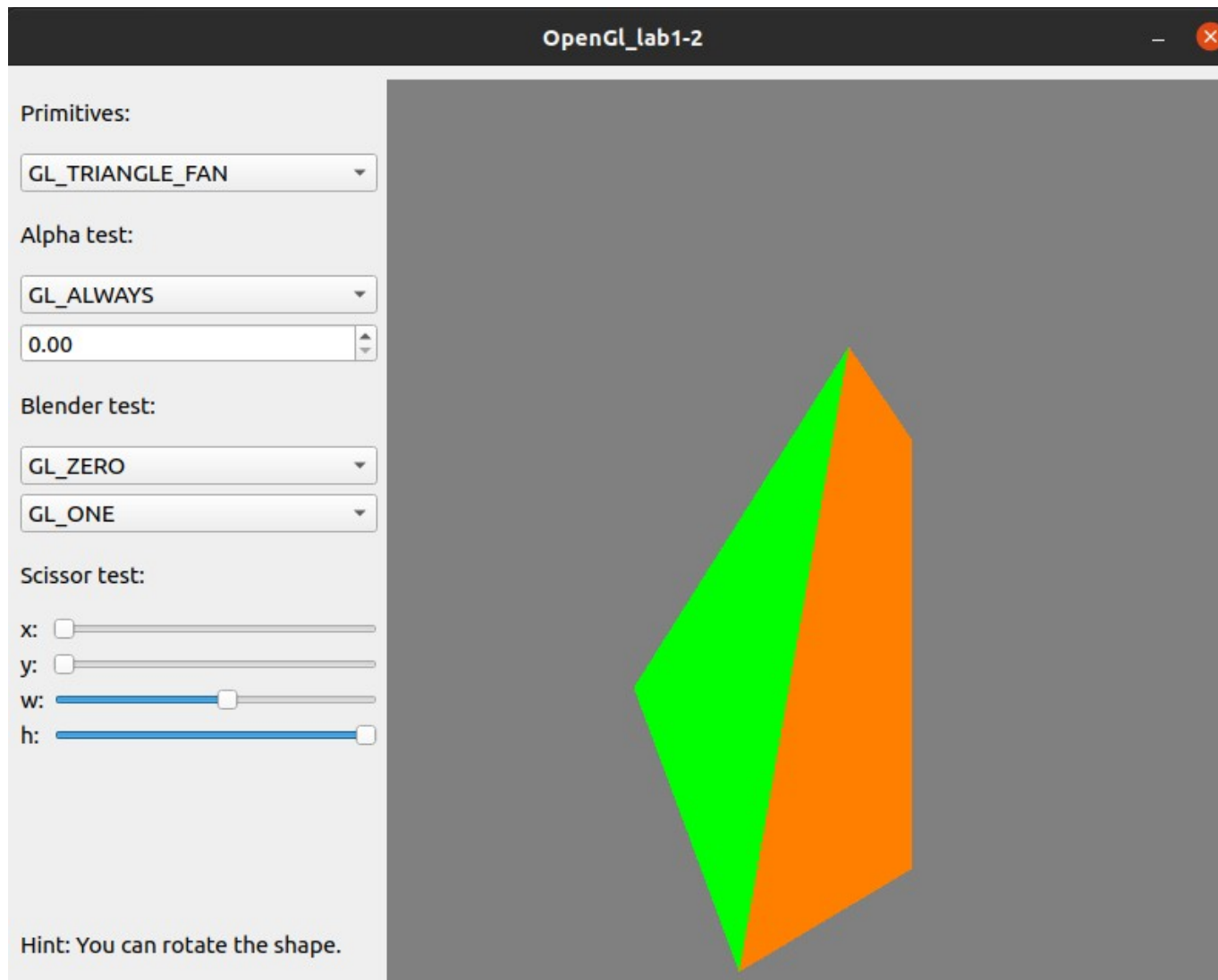
Тест смешивания цветов



Тест прозрачности



Тест 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.