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 _____ Ларин А.

Преподаватель _____ Герасимова Т.В.

Санкт-Петербург

2021

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 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

Задание вполнено на языке C++ с использованием фреймворка Qt5.
Qt предоставляет возможность использование OpenGL посредством класса QGLWidget. От него отнаследован класс CustomGL.
`class CustomGL : public QGLWidget, в котором переопределены методы:`

- `void initializeGL();`
  Задает контекст рендеринга. Заключается в установке фона белым
- `void resizeGL(int nWidth, int nHeight);`
  `Для корректного масштабирования. Размер окна задается через glViewport.`
- `void paintGL();`
  Вызов этой функции происходит при рисовании. Она в свою очередь вызывает функцию `scene` где прописана вся логика рисования.

В данном классе так же прописаны шесть точек. Для них выбраны различные цвета, а так при отрисовке задается различная прозрачность. Точки расположены в форме звезды.

В функции scene происходит отрисовка примитива не экране. Последовательно вызываются функции glEnable с параметрами GL_BLEND, GL_SCISSOR_TEST, GL_ALPHA_TEST и функции задающие параметры для них:
Для GL_BLEND: sfactor и dfactor
Для GL_SCISSOR_TEST: координаты обрезки
Для GL_ALPHA_TEST: тип и пороговое значение

Все параметры берутся из главного окна посредством сигналов при изменении

В коде главного окна инициализируются элементы интерфейса, помещаются в layout'ы и связываются сигналы со слотами.

Выводы.

Были изучены основы работы с OpenGL, отрисованы примитивы и произведены эксперименты со смешиванием, обрезкой и прозрачностью