

Программные средства САПР
Задания на последнюю лабораторную работу по OpenGL №2 (ЭН-34)

В этой лабораторной работе вы должны построить 3D фигуру на основе файлов – данных технологического моделирования.

В качестве примеров для вас подготовлены существенно упрощённые файлы, описывающие сетку трёхмерного объекта. Основой для подготовленных для вас примеров являются файлы настоящего САПР технологического моделирования Synopsys TCAD в формате DF-ISE.

В формате DF-ISE данные сетки хранятся в файле с расширением .grd, который устроен следующим образом (с учётом сделанного для вас упрощения).

Все данные разделены на секции, названия секций соответствуют тому, что в этих секциях описывается. Всего в файле присутствует три вида секций.

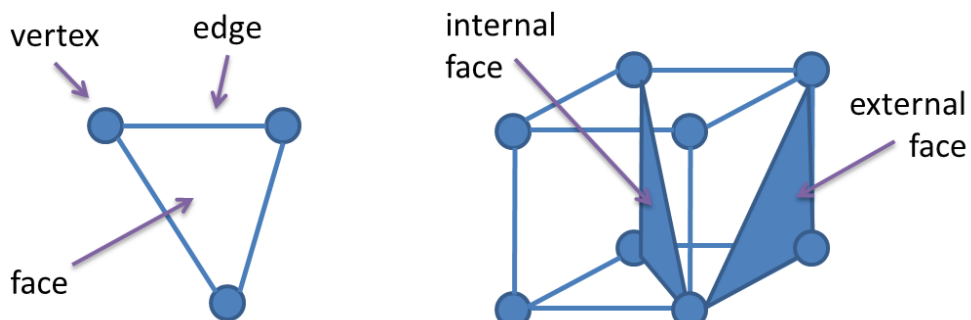
1. **Vertices** – это массив координат вершин точек-узлов сетки.
2. **Edges** – массив рёбер сетки. Для каждого ребра хранятся индексы вершин из массива **Vertices**, которые в него входят.
3. **Faces** – массив поверхностей. Первым идёт ключевое слово, иллюстрирующее, какого типа поверхность: треугольник или четырёхугольник. В вашем примере все поверхности – треугольники, в каждой строчке есть три числа, число показывает номер ребра из массива **Edges**, образующее треугольник. Если число отрицательное, это значит, что ребро переориентируется, то есть вместо того, чтобы быть нарисованным от 1 к 2 точке, ребро рисуется от 2 к 1 точке и его индекс берётся положительным и на единицу меньше.

Например, в файле cube.grd в разделе Faces есть такая поверхность: «-14 2 7». Это значит, что первым берётся ребро 13, а его начальная и конечная точки меняются местами, остальные рёбра (2 и 7) рисуются так, как и должны.

Если в файле встречается поверхность не треугольник, вы должны выдать в консоль предупреждение, что встретилась неподдерживаемая поверхность, указать номер строки раздела Faces, на которой встретился такой элемент, и проигнорировать его.

4. **Locations** – массив, показывающий положение соответствующей поверхности из массива Faces: 'i' означает, что поверхность располагается внутри фигуры, а 'e' – что снаружи.

Пример взаимного отношения элементов и их расположение показано на рисунке ниже.



В процессе выполнения данной лабораторной работы у вас есть возможность получить три плюса за следующие достижения:

1. написать программу, которая считывает данные из файла и выводит в консоль информацию о найденных максимальном и минимальном значениях по всем осям;
2. нарисовать при помощи библиотеки OpenGL фигуру;
3. дополнить код небольшим индивидуальным заданием.

Что нужно сделать обязательно всем вариантам:

1. фигура должна рисоваться по центру экрана и вращаться относительно центра;
2. приближение/удаление – на клавиши «+»/«-»;
3. клавиша Esc – выход из программы.

Код для работы с сеткой устройства, точки которого читаются из файла, должен быть реализован в виде класса.

Класс должен иметь интерфейсные функции:

- функция для чтения данных из файла, возвращает bool, чтобы было понятно, удалось прочесть файл или нет;
- функция для рисования фигуры;
- функция для реализации индивидуального задания.

Пример того, как должен выглядеть ваш код для работы с сеткой устройства, приведён ниже.

```
class Grid {
    std::vector<Vertex> vertices;
    std::vector<Edge> edges;
    std::vector<Face> faces;
public:
    bool Load(std::string fileName);
    void Draw();
    ...
};
```

ВАРИАНТЫ

№	Задания
1 5 9 13 17 21	Фон окна – светло-серый. Фигура – тёмно-зелёная. Сетка фигуры – светло-синяя. При нажатии на клавишу «М» должен меняться режим отрисовки фигуры: вместо заливки рисуется массив внешних рёбер (рёбер внешних поверхностей) и наоборот. Рёбра рисуются удвоенной толщиной линии.
2 6 10 14 18 22	Фон окна – светло-серый. Фигура – чёрная. Сетка фигуры – белая. При нажатии на клавишу «М» должен меняться перечень рисуемых поверхностей. Рисоваться должны все / только внутренние поверхности. Клавишей «Enter» включается/выключается отрисовка фигуры по точкам. Сетка рисуется всегда.
3 7 11 15 19 23	Фон окна – тёмно-серый. Фигура – светло-серая. Сетка фигуры – тёмно-синяя. При нажатии на клавишу «М» должен меняться режим отрисовки фигуры: вместо заливки рисуется массив внутренних рёбер (рёбер внутренних поверхностей) и наоборот. Рёбра рисуются удвоенной толщиной линии.
4 8 12 16 20 24	Фон окна – тёмно-серый. Фигура – белая. Сетка фигуры – тёмно-синяя. При нажатии на клавишу «М» должен меняться перечень рисуемых поверхностей. Рисоваться должны все / только внешние поверхности. Клавишей «Enter» включается/выключается отрисовка фигуры по точкам. Сетка рисуется всегда.

Дополнительные сведения для выполнения лабораторной работы.

Q: Как поменять размер пикселя при отрисовке по точкам?

A: Для этого нужно воспользоваться функцией **glPointSize**, имеющей следующий формат:

```
void glPointSize(GLfloat size)
```

Q: Как поменять толщину линии при отрисовке линиями?

A: Для этого нужно воспользоваться функцией **glLineWidth**, имеющей следующий формат:

```
void glLineWidth(GLfloat size)
```

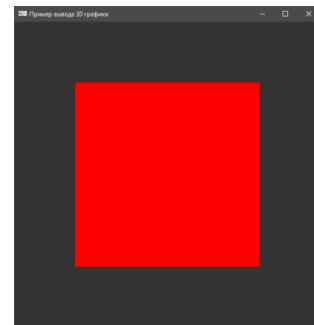
Q: Как нарисовать сетку поверх фигуры?

A: Для того, чтобы нарисовать сетку, используется следующий подход (показываю на примере рисования группы смежных клеток). Сначала рисуется обычная фигура.

Код

```
glColor3f(1, 0, 0);
glBegin(GL_QUADS);
for (float y = -5.0; y < 5.0; y += 1)
    for (float x = -5.0; x < 5.0; x += 1) {
        glVertex3f(x, y, 0.0f);
        glVertex3f(x + 1, y, 0.0f);
        glVertex3f(x + 1, y + 1, 0.0f);
        glVertex3f(x, y + 1, 0.0f);
    }
glEnd();
```

Результат



Затем фигура перерисовывается (на место, а после!), при этом устанавливается способ рисования поверхностей линиями, задаётся новый цвет и указывается смещение (серый код писать нужно, но он точно такой же, как и пример выше).

Код

```
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
glLineWidth(1);
glEnable(GL_POLYGON_OFFSET_LINE);
glPolygonOffset(-1, -1);
glColor3f(1, 0, 0);
glBegin(GL_QUADS);
for (float y = -5.0; y < 5.0; y += 1)
    for (float x = -5.0; x < 5.0; x += 1) {
        glVertex3f(x, y, 0.0f);
        glVertex3f(x + 1, y, 0.0f);
        glVertex3f(x + 1, y + 1, 0.0f);
        glVertex3f(x, y + 1, 0.0f);
    }
glEnd();
glDisable(GL_POLYGON_OFFSET_LINE);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
```

Результат

