

**Программные средства САПР**  
**Задания на лабораторную работу по OpenGL (ЭН-34)**

В файле задаётся матрица из чисел с плавающей точкой, размер матрицы всегда равен 101x101 элемент (кроме примера test\_38\_00.txt, там размер 4x4).

Задание на лабораторную работу: визуализировать с помощью OpenGL поверхности, заданные матрицами во входных файлах.

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;
2. нарисовать при помощи библиотеки OpenGL поверхность, добавив код для вращения вокруг всех осей (реализацию вращения можно сделать как угодно - можно вращать с клавиатуры, можно мышкой, по вашему усмотрению);
3. дополнить код индивидуальным заданием.

Что нужно сделать обязательно всем вариантам:

1. фигура должна рисоваться по центру экрана и вращаться относительно центра;
2. зная, что размер нашей матрицы 101x101, сделайте так, чтобы фигура во время запуска сразу вся была видна на экране (максимальный размах по оси Z составляет от -5 до +5);
3. приближение/удаление – на клавиши «+»/«-»;
4. отрисовку сделать в режиме сетки (каркасная модель), чтобы было лучше видно, что рисуется;
5. клавиша Esc – выход из программы.

**Варианты**

№	Задания
1, 7, 13, 19	Управляющими клавишами – стрелками «вверх» и «вниз» - реализовать сжатие и растяжение фигуры вдоль оси Z. Цвет фона медленно меняется от серого до чёрного и обратно. Цвет отрисовки сетки – жёлтый.
2, 8, 14, 20	В размер фигуры рисуется плоскость, перпендикулярная оси Z в точке Z = 0. При нажатии на клавишу «Enter» эта плоскость перемещается на максимум, при нажатии – на минимум, при нажатии – в центр. Цвет фона медленно меняется от белого до жёлтого и обратно. Цвет отрисовки сетки – синий.
3, 9, 15, 21	Фигура постоянно медленно вращается вокруг оси Z. При нажатии на клавишу «Пробел» можно останавливать и снова запускать это вращение. Клавишами «Вверх» и «Вниз» можно менять скорость. Цвет фона медленно меняется от светло-синего до белого и обратно. Цвет отрисовки сетки – тёмно-зелёный.
4, 10, 16, 22	При нажатии на клавишу «A» включать/выключать оси, которые рисуются разными цветами. Рисовать последовательно: рисуется только X, только Y, только Z, все вместе. Цвет фона медленно меняется от серого до чёрного и обратно. Цвет отрисовки сетки – светло-серый.

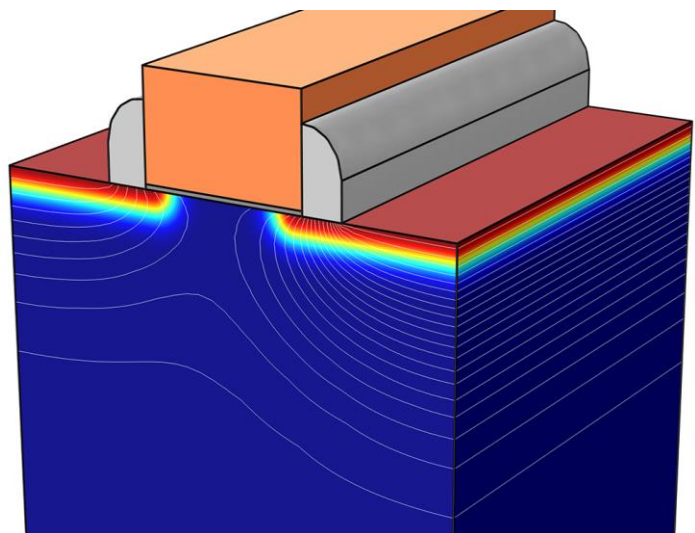
5, 11, 17, 23	При нажатии на клавишу «М» нужно рисовать точки, имеющие значения, большие, чем 0, размером 5 единиц. Цвет – по выбору. Цвет фона медленно меняется от светло-серого до белого и обратно. Цвет отрисовки сетки – красный.
6, 12, 18, 24	На поверхности в точке $x, y, z$ рисуется точка размером в 5 единиц. При нажатии на клавиши стрелок эту точку можно перемещать по сетке. Цвет фона медленно меняется от зелёного до чёрного и обратно. Цвет отрисовки сетки – красный.

Само собой, сделать с применением ООП.

Класс сетки имеет функции для чтения, отрисовки, смены состояния (реакция на клавиши).

#### **Задание повышенной сложности.**

Цвета рисовать по правилам отрисовки технологических САПР: синий – минимальное значение, красный – максимальное, цвета меняются так: синий-зелёный-жёлтый-красный (см. рисунок справа);



## 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 реализовать вращение?

**A:** Вращение проще всего реализовать путём задания функции, которая будет вызываться каждый раз, когда ничего не происходит. Эта функция возвращает `void` и не принимает аргументов, связывается с библиотекой OpenGL путём передачи её имени в функцию `glutIdleFunc` в функции `main`.

*Пример:*

```
void idle() {
    // тут меняется угол
    glutPostRedisplay();
}

void main(int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA);
    glutInitWindowPosition(100,100);
    glutInitWindowSize(800,800);
    glutCreateWindow("3D");
    glEnable(GL_DEPTH_TEST);
    glutKeyboardFunc(readKB);
    glutDisplayFunc(renderScene);

    glutIdleFunc(idle);

    glutMainLoop();
    return 0;
}
```

**Q:** Как считать нажатие специальных клавиш (стрелки, функциональные)?

**A:** Для этого используется механизм, аналогичный чтению обычных клавиш, разница лишь в том, что:

1. функция принимает 3 аргумента, но первый – не `unsigned char`, а `int` – это тот же код клавиши;
2. функция привязывается к библиотеке OpenGL не с помощью `glutKeyboardFunc`, а `glutSpecialFunc`.

*Пример:*

```
void readSK(int pressKey, int x = 0, int y = 0){
    switch(pressKey) {
        case GLUT_KEY_RIGHT : printf("right pressed\n");
                             break;
    }
    glutPostRedisplay();
}

void main(int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA);
    glutInitWindowPosition(100,100);
    glutInitWindowSize(800,800);
    glutCreateWindow("3D");
}
```

```

glEnable(GL_DEPTH_TEST);
glutDisplayFunc(renderScene);

glutSpecialFunc(readSK);

glutMainLoop();
return 0;
}

```

### Q: Как ловить события мыши?

A: Раздельно ловятся факты нажатия и перемещения мыши. Нажатие ловится функцией, связываемой с OpenGL посредством `glutMouseFunc`.

*Пример:*

```

void mouse(int button, int state, int x, int y) {
    if(button == GLUT_LEFT_BUTTON)
        printf("left button!");
    if(state == GLUT_UP)
        printf("button up!");
}

void main(int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA);
    glutInitWindowPosition(100,100);
    glutInitWindowSize(800,800);
    glutCreateWindow("3D");
    glEnable(GL_DEPTH_TEST);
    glutDisplayFunc(renderScene);

    glutMouseFunc(mouse);

    glutMainLoop();
    return 0;
}

```

Перемещение ловится с помощью функции, которая связывается с OpenGL с помощью `glutMotionFunc`. Эта функция вызывается только тогда, когда перемещение происходит с нажатой клавишей, поэтому факт нажатия можно не проверять.

*Пример:*

```

void motion(int x, int y) {
    printf("Yahoo! I've been moved!\n");
}

void main(int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA);
    glutInitWindowPosition(100,100);
    glutInitWindowSize(800,800);
    glutCreateWindow("WaveViewer");
    glEnable(GL_DEPTH_TEST);
    glutDisplayFunc(renderScene);

    glutMotionFunc(motion);

    glutMainLoop();
    return 0;
}

```