

Лабораторная работа №4 «Поворот. Перенос. Масштаб»

Для изменения масштаба используется команда `glScalef(x,y,z)` с тремя аргументами, представляющими собой масштабные множители для каждой из осей. Например, если перед командными скобками вставить строку

```
glScalef(0.5, 0.5, 1.0);
```

то будет нарисована уменьшенная в два раза фигура. После отрисовки необходимо обязательно вернуть масштаб к исходному, т.е. в данном примере добавить строку т. е в данном случае добавить строку:

```
glScalef(2.0, 2.0, 1.0);
```

Например, нарисуем квадрат, который занимает всю область экрана, однако за счет применения команды масштабирования он будет в 2 раза меньше.

```
glScalef(0.5, 0.5, 1.0);
glBegin(GL_QUADS);
    glVertex2f(1,1);
    glVertex2f(1,-1);
    glVertex2f(-1,-1);
    glVertex2f(-1,1);
glEnd;
glScalef(2.0, 2.0, 1.0);
```

Восстанавливать масштаб необходимо для того, чтобы каждое последующее обращение к обработчику перерисовки экрана не приводило бы к последовательному уменьшению/увеличению изображения. Масштабные множители могут иметь отрицательные значения, при этом изображение переворачивается по соответствующей оси. При двумерных построениях значение коэффициента по оси Z безразлично.

Для поворота используется команда `glRotatef(alpha, x, y, z)` с четырьмя аргументами: угол поворота в градусах, и вектор поворота – три вещественных числа. Для двумерных построений наиболее нагляден поворот по оси Z.

При положительном значении компоненты вектора поворот осуществляется против часовой стрелки, при отрицательном – по часовой стрелке. Важно не само значение компоненты, а ее знак и равенство/неравенство ее нулю.

```
glRotatef(5, 0.0, 0.0, 1.0);
glBegin(GL_QUADS);
    glVertex2f(0.5, 0.5);
    glVertex2f(0.5, -0.5);
    glVertex2f(-0.5, -0.5);
    glVertex2f(-0.5, 0.5);
glEnd;
```

На сцене появится квадрат, повернутый на 5 градусов.

Если к этому добавить обработчик `onKeyDown`, с одной командой `Refresh`, то при нажатии любой кнопки квадрат будет поворачиваться на 5 градусов от текущего состояния. Точно так же, как было с масштабом, поворот действует на все последующие команды воспроизведения, так что при необходимости текущее состояние восстанавливается обратным поворотом.

```
glRotatef(45, 0.0, 0.0, 1.0);
glBegin(GL_POLYGON);
    glVertex2f(-0.6, -0.1);
    glVertex2f(-0.6, 0.4);
    glVertex2f(-0.1, 0.4);
    glVertex2f(-0.1, -0.1);
    glColor3f(random, random, random);
    glLineStipple(random(5), random($FFFF));
    glBegin(GL_LINES);
        glVertex2f(xpos, ypos);
        glVertex2f(xpos+0.5*random*sin(random(360)), ypos+0.5*random*cos(random(360)));
    glEnd;
end;
```