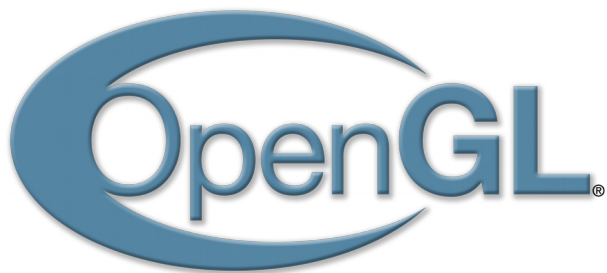


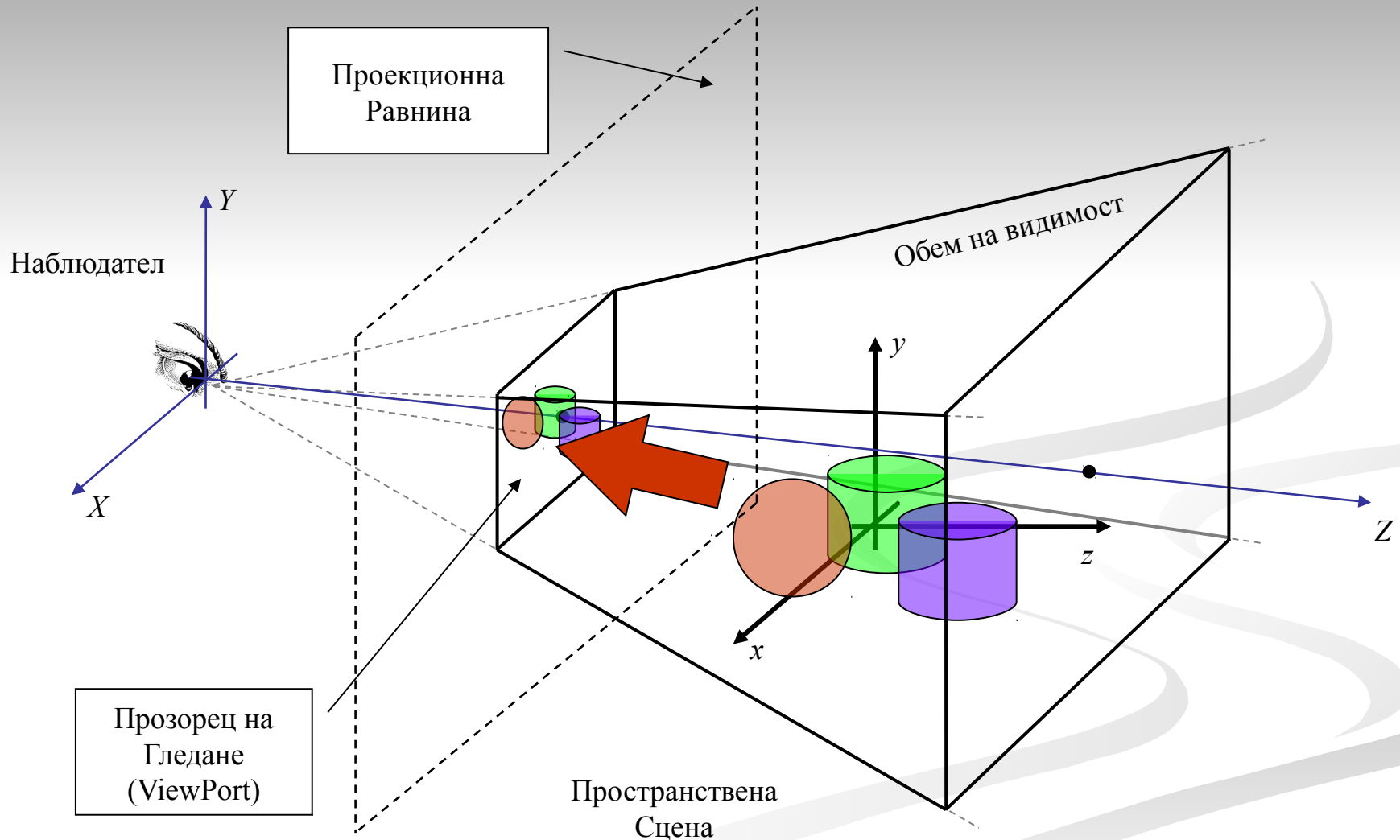


OpenGL

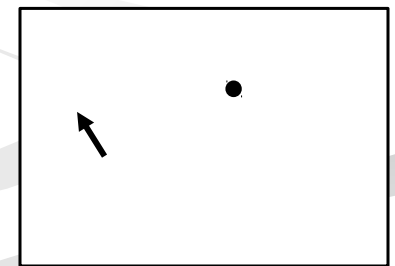
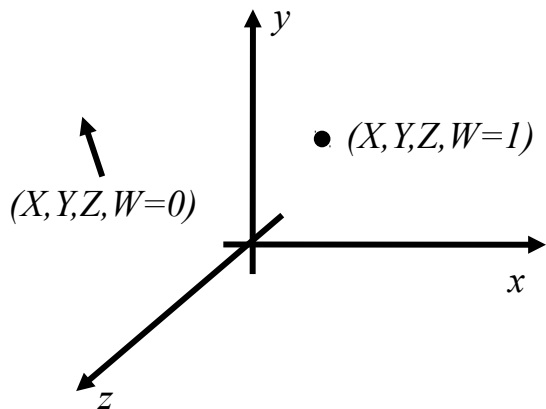
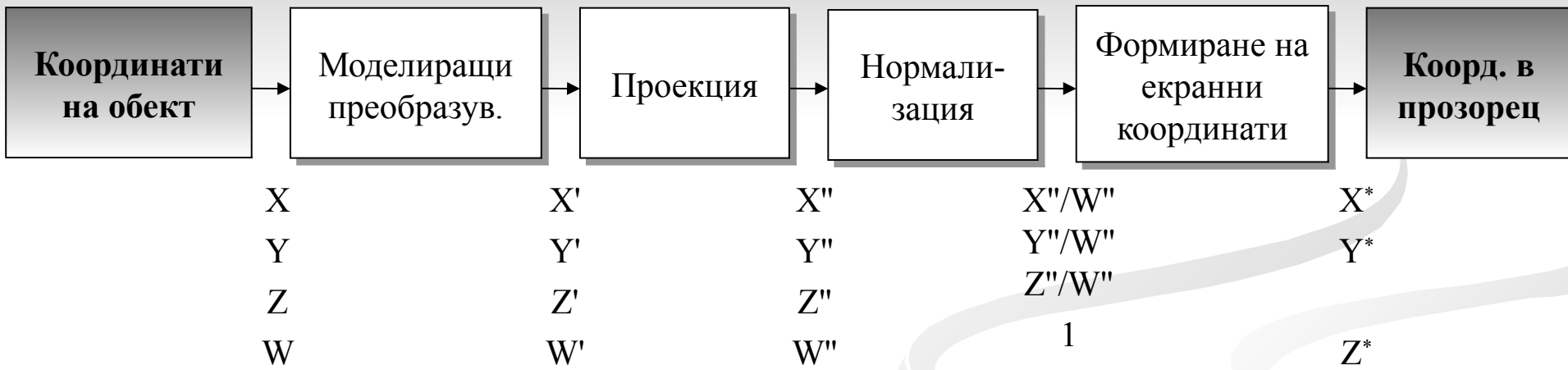
Трансформации



Обща Постановка



Преобразуване на Координати



Трансформации в OpenGL

- В OpenGL трансформациите се извършват чрез операции с матрици;
- Има няколко стека с матрици служещи за различни видове трансформации;
- В даден момент може да работим само с един от тези стекове;
- От всички тях се ползва само матрицата на върха на дадения стек.

Избор на активен стек

```
void glMatrixMode (GLenum mode)
```

mode :

| | |
|----------------------------|---------------------------|
| <code>GL_MODELVIEW</code> | Модела |
| <code>GL_PROJECTION</code> | Проекция |
| <code>GL_TEXTURE</code> | Текстурните координати |
| <code>GL_COLOR</code> | Цвета |

Пример :

```
glMatrixMode (GL_PROJECTION) ;
```

Зареждане на Матрици

```
void glLoadIdentity(void)
void glLoadMatrix... (*m)
void glLoadTransposeMatrix... (*m)
```

Пример:

```
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glMatrixMode(GL_MODELVIEW);
glLoadMatrixd(M);
```

glLoadIdentity()

| |
|-------|
| M_4 |
| M_3 |
| M_2 |
| M_1 |



| |
|----------|
| I |
| M_3 |
| M_2 |
| M_1 |

glLoadMatrix...(M)

| |
|-------|
| M_4 |
| M_3 |
| M_2 |
| M_1 |



| |
|----------|
| M |
| M_3 |
| M_2 |
| M_1 |

glLoadTransposeMatrix...(M)

| |
|-------|
| M_4 |
| M_3 |
| M_2 |
| M_1 |



| |
|-------|
| M^T |
| M_3 |
| M_2 |
| M_1 |

Умножение на Матрици

```
void glMultMatrix... (*m) ;  
void glMultTransposeMatrix... (*m) ;
```

Пример :

```
glLoadMatrixd (M1) ;  
glBegin (GL_POINTS) ;  
    glVertex3d (1, 2, 3) ;  
glEnd () ;  
glMultMatrixd (M2) ;  
glBegin (GL_POINTS) ;  
    glVertex3d (1, 2, 3) ;  
glEnd () ;
```

glMultMatrix...(M)

| |
|-------|
| M_4 |
| M_3 |
| M_2 |
| M_1 |



| |
|-----------|
| $M_4 * M$ |
| M_3 |
| M_2 |
| M_1 |

glMultTransposeMatrix...(M)

| |
|-------|
| M_4 |
| M_3 |
| M_2 |
| M_1 |



| |
|-------------|
| $M_4 * M^T$ |
| M_3 |
| M_2 |
| M_1 |

Операции над Стека

```
void glPushMatrix(void)
void glPopMatrix(void)
```

Пример :

```
glPushMatrix();
glMultMatrixd(M1);
glBegin(GL_POINTS);
    glVertex3d(1,2,3);
glEnd();
glPopMatrix();
```

glPushMatrix()

| |
|-------|
| M_4 |
| M_3 |
| M_2 |
| M_1 |



| |
|-------|
| M_4 |
| M_4 |
| M_3 |
| M_2 |
| M_1 |

glPopMatrix()

| |
|-------|
| M_4 |
| M_3 |
| M_2 |
| M_1 |



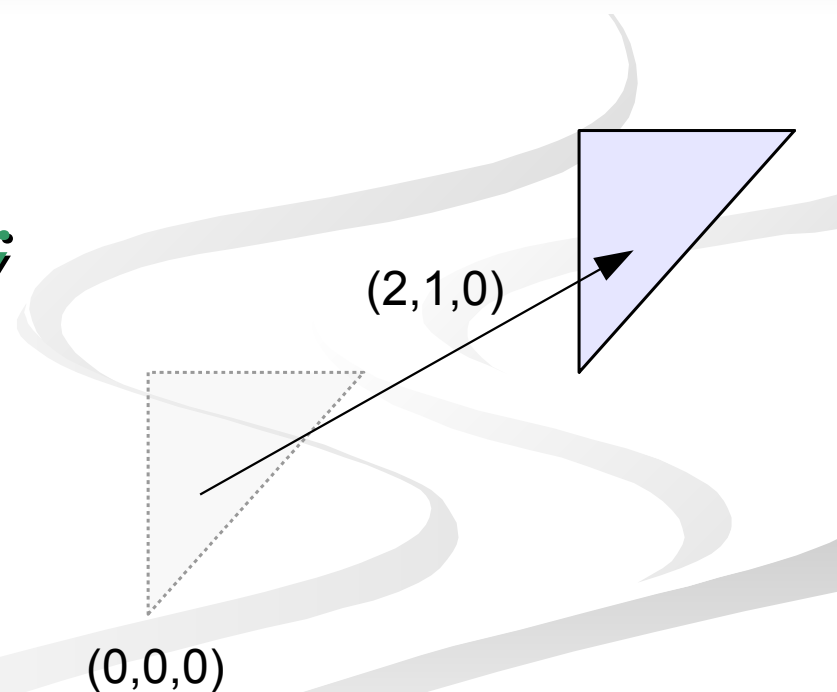
| |
|-------|
| M_3 |
| M_2 |
| M_1 |

Транслация

```
void glTranslate... (dx, dy, dz)
```

Пример:

```
glTranslated(2, 1, 0);  
glBegin(GL_TRIANGLES);  
    glVertex3d(0, 0, 0);  
    glVertex3d(0, 1, 0);  
    glVertex3d(1, 1, 0);  
glEnd();
```



glTranslate...(dx,dy,dz)

| |
|-------|
| M_4 |
| M_3 |
| M_2 |
| M_1 |



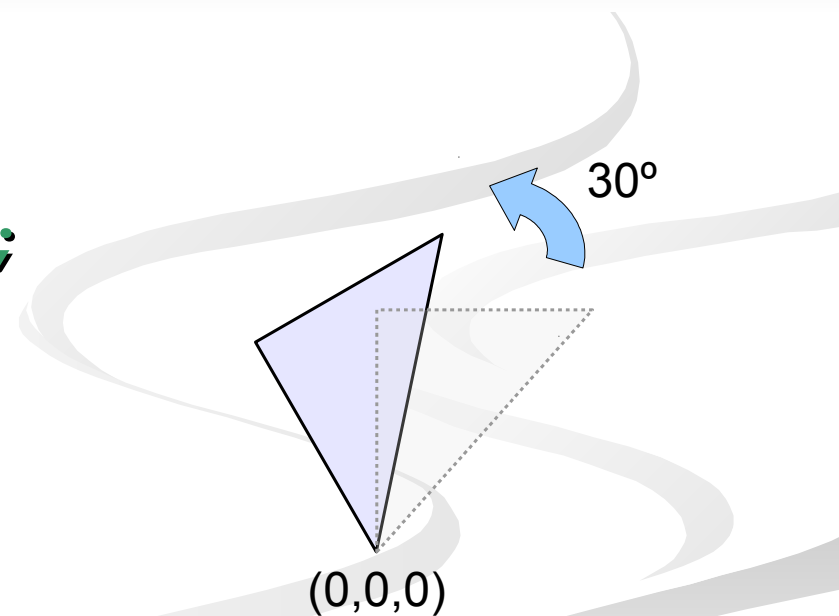
| |
|---------------------|
| $M_4 * T(dx,dy,dz)$ |
| M_3 |
| M_2 |
| M_1 |

Ротация

```
void glRotate... (angle, vx, vy, vz)
```

Пример:

```
glRotated(30, 0, 0, 1);  
glBegin(GL_TRIANGLES);  
    glVertex3d(0, 0, 0);  
    glVertex3d(0, 1, 0);  
    glVertex3d(1, 1, 0);  
glEnd();
```



glRotate...(a, vx,vy,vz)

| |
|-------|
| M_4 |
| M_3 |
| M_2 |
| M_1 |



| |
|--------------------------|
| $M_4 * R(a, vx, vy, vz)$ |
| M_3 |
| M_2 |
| M_1 |

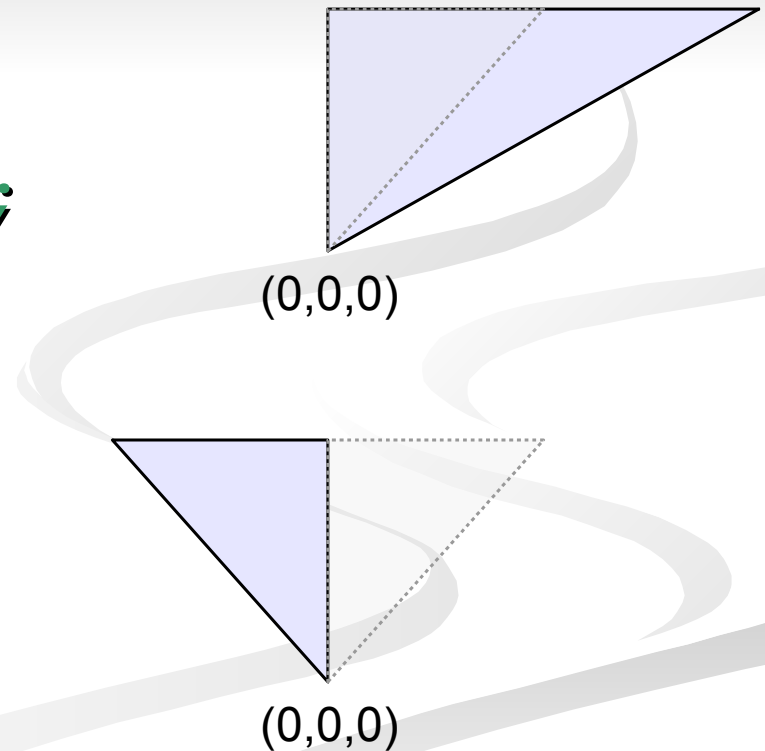
Мащабиране

```
void glScale... (sx, sy, sz)
```

Пример:

```
glScaled(2, 1, 1);  
glBegin(GL_TRIANGLES);  
    glVertex3d(0, 0, 0);  
    glVertex3d(0, 1, 0);  
    glVertex3d(1, 1, 0);  
glEnd();
```

```
glScaled(-1, 1, 1); // огледално по X
```



glScale...(sx,sy,sz)

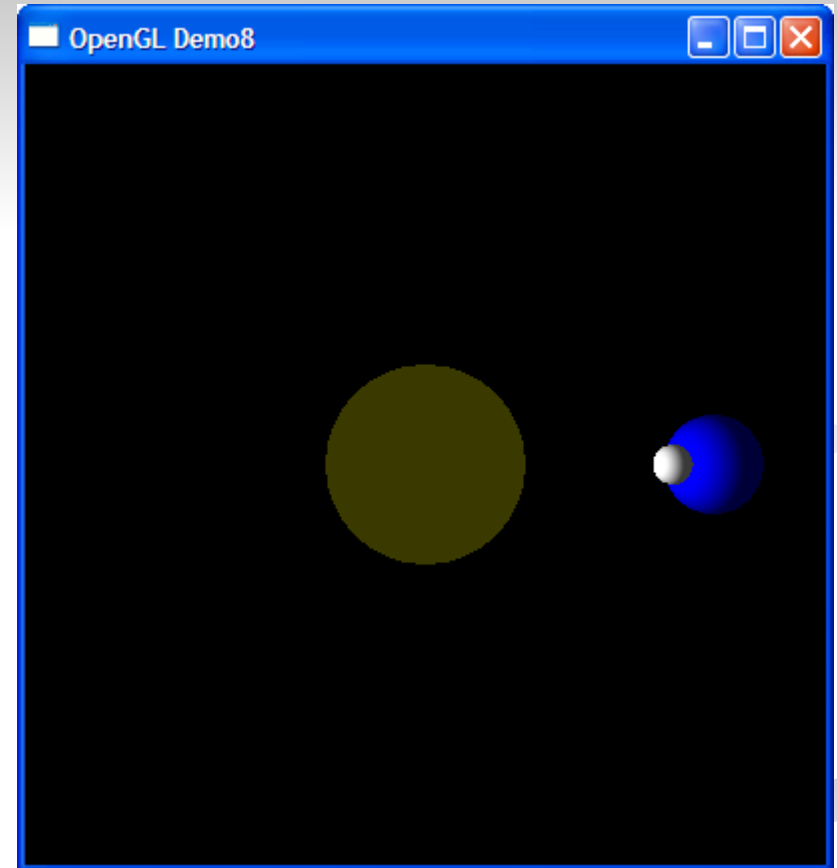
| |
|-------|
| M_4 |
| M_3 |
| M_2 |
| M_1 |



| |
|---------------------|
| $M_4 * S(sx,sy,sz)$ |
| M_3 |
| M_2 |
| M_1 |

Пример 1

```
...  
glLoadIdentity();  
glRotated(angle1, 0,1,0);  
glColor3d(1,1,0);  
glutSolidSphere(0.5,32,32);  
  
glTranslated(2,0,0);  
glRotated(angle2, 0,1,0);  
glColor3d(0,0,1);  
glutSolidSphere(0.25,32,32);  
  
glTranslated(0.5,0,0);  
glColor3d(1,1,1);  
glutSolidSphere(0.1,32,32);  
...
```



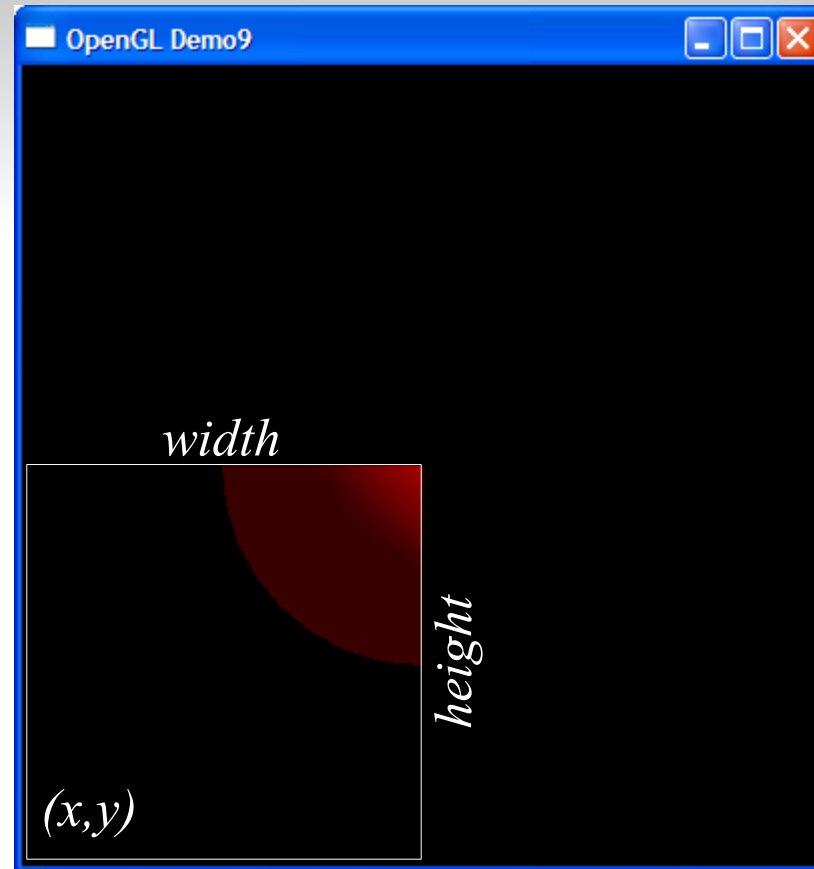
Изрязване

```
void glScissor(GLint x, GLint y,  
              GLsizei width, GLsizei height)  
glEnable(GL_SCISSOR_TEST);
```

Пример:

```
glEnable(GL_SCISSOR_TEST);  
glScissor(0, 0, 200, 200);  
glColor3d(1, 0, 0);  
glutSolidSphere(1, 32, 32);
```

Пример 2



Други “Режещи инструменти”

```
void glClipPlane(GLenum plane,  
    const GLdouble *equation)
```

```
void glGetClipPlane(GLenum plane,  
    GLdouble *equation)
```

```
glEnable(GL_CLIP_PLANE0)
```

Пример:

```
GLdouble plane[4] = {1,2,3,4};  
glClipPlane(GL_CLIP_PLANE0, plane);  
glutSolidSphere(1,32,32);
```

Пример 3 (1/2)

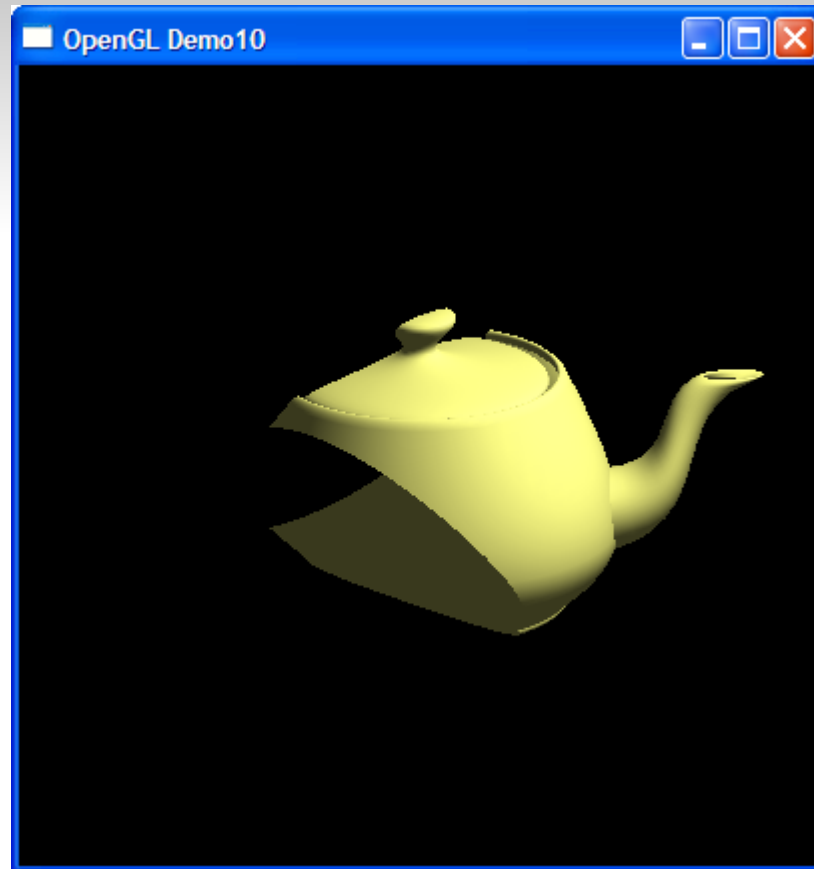
```
GLdouble plane0[4] = {1,1,-1,0.75};  
GLdouble plane1[4] = {1,-1,1,0.75};
```

...

```
glEnable(GL_CLIP_PLANE0);  
glEnable(GL_CLIP_PLANE1);  
glClipPlane(GL_CLIP_PLANE0, plane0);  
glClipPlane(GL_CLIP_PLANE1, plane1);
```

```
glColor3d(1,1,0.5);  
glutSolidTeapot(1);
```

Пример 3 (2/2)



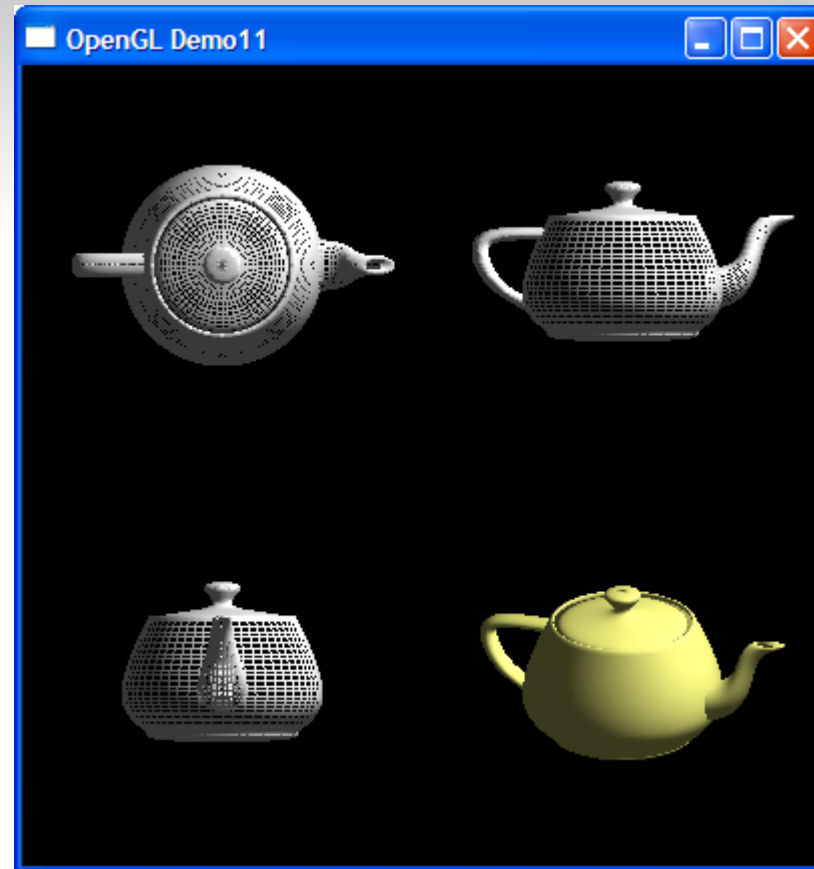
Прозорец на гледане

```
void glViewport(  
    GLint x, GLint y,  
    GLsizei width, GLsizei height)
```

Пример :

```
glViewport(0,0,200,200);  
glColor3d(1,1,1);  
glutSolidTeapot(1);  
glViewport(200,200,400,400);  
glutSolidTeapot(1);
```

Пример 4

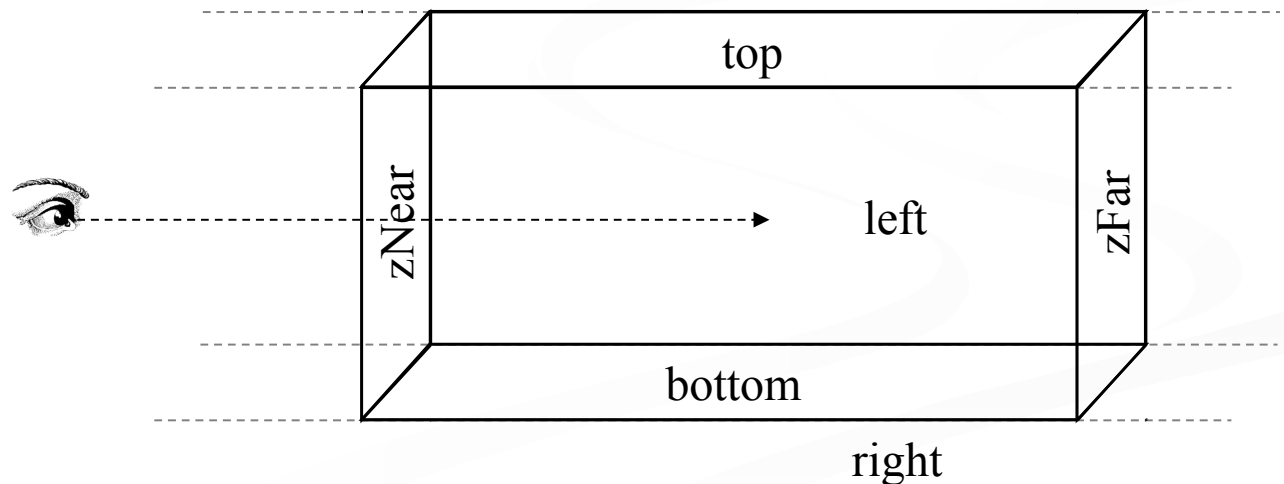


Проекции – Ортогонална

```
void glOrtho(  
    GLdouble left, GLdouble right,  
    GLdouble bottom, GLdouble top,  
    GLdouble zNear, GLdouble zFar)
```

Пример:

```
glOrtho(  
    -1, 1,  
    -1, 1,  
    1, 5);
```

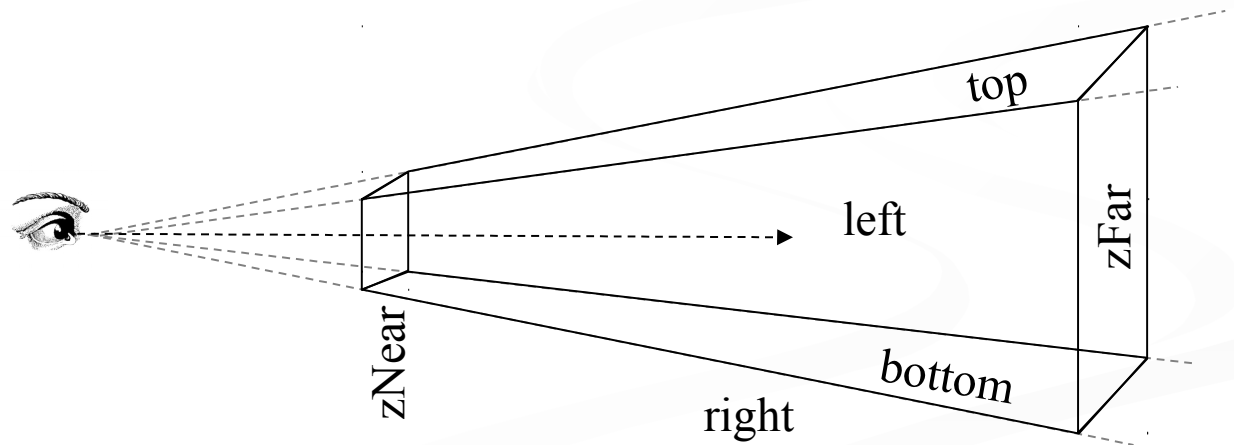


Проекции – Перспективна

```
void glFrustum(  
    GLdouble left, GLdouble right,  
    GLdouble bottom, GLdouble top,  
    GLdouble zNear, GLdouble zFar)
```

Пример :

```
glFrustum(  
    -1, 1,  
    -1, 1,  
    1, 5);
```



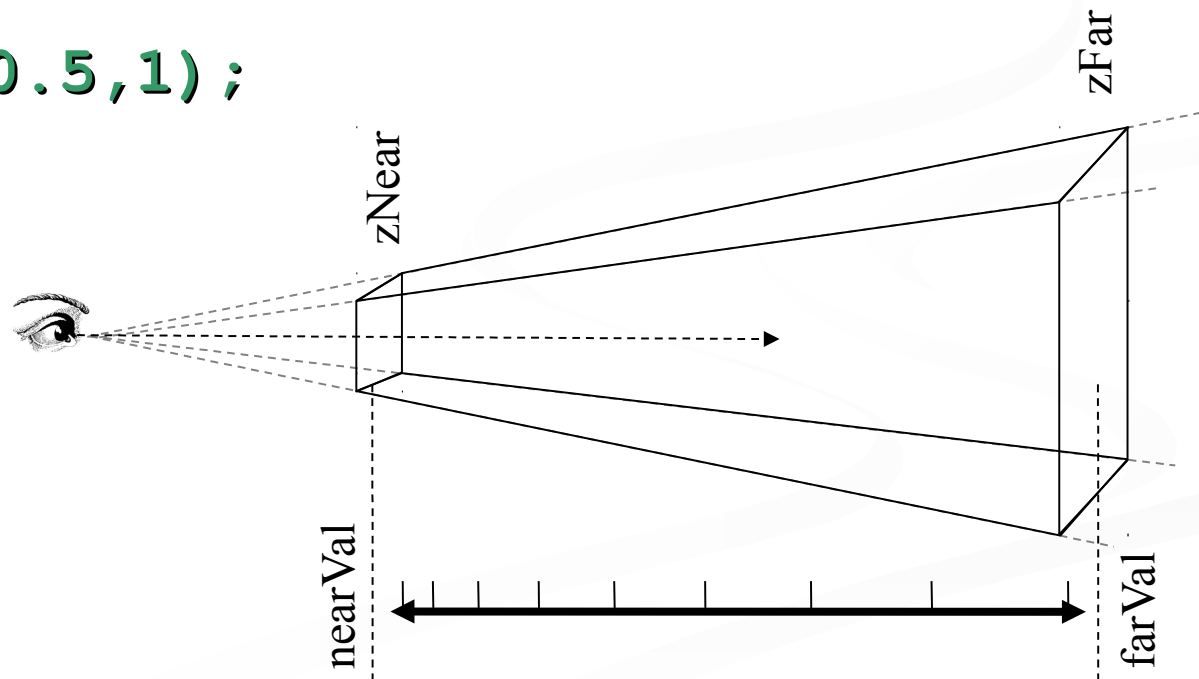
Нормиране на дълбочината

```
void glDepthRange(  
    GLclampd nearVal, GLclampd farVal)
```

Пример:

```
glDepthRange(0.5, 1);
```

...

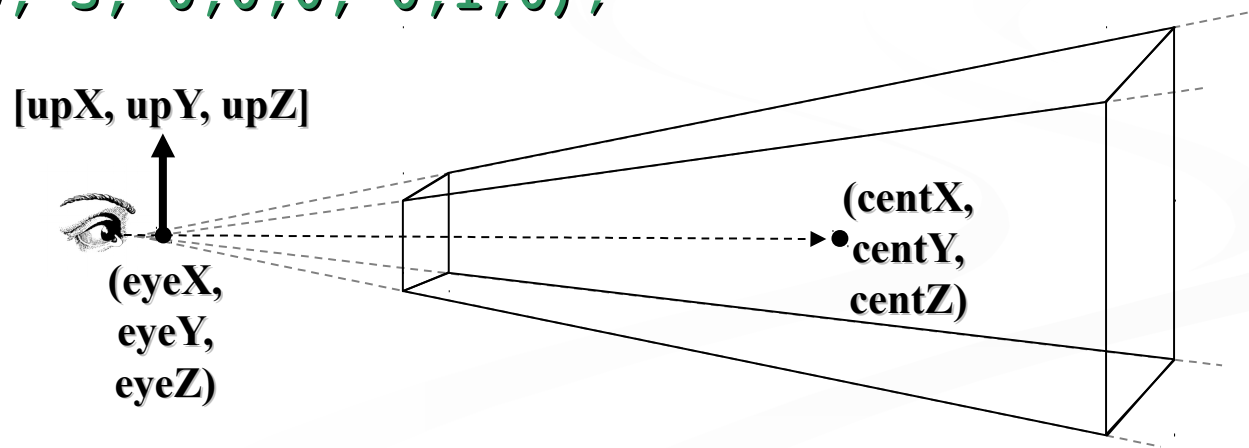


Помощни функции - gluLookAt

```
void gluLookAt(  
    GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ,  
    GLdouble centX, GLdouble centY, GLdouble centZ,  
    GLdouble upX, GLdouble upY, GLdouble upZ)
```

Пример:

```
gluLookAt(0,0,-5, 0,0,0, 0,1,0);
```



Помощни функции - gluPerspective

```
void gluPerspective(  
    GLdouble fovy, GLdouble aspect,  
    GLdouble zNear, GLdouble zFar)
```

Пример:

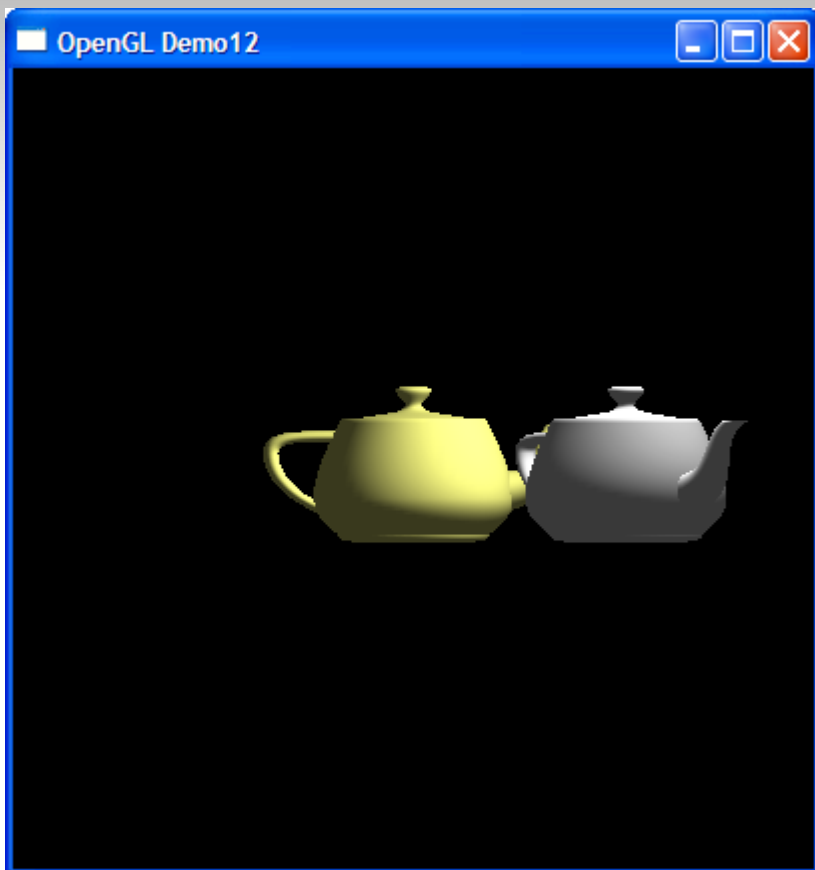
```
gluPerspective(60, w/h, 0, 1);
```

Пример 5 (1/2)

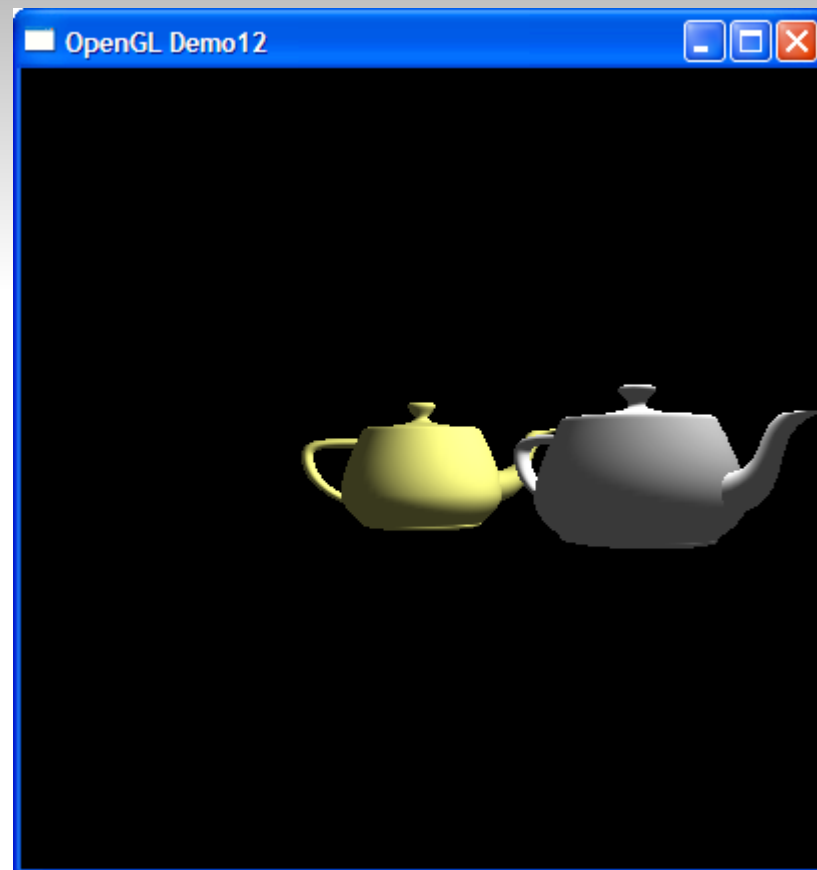
```
...
glLoadIdentity();
glColor3d(1,1,0.5);
glutSolidTeapot(1);
glRotated(angle, 0,1,0);
glTranslated(3,0,0);
glColor3d(1,1,1);
glutSolidTeapot(1);
...

// В resize()
glOrtho(-4,4, -4,4, 2,20);
// или
// glFrustum(-1,1, -1,1, 2,20);
```

Пример 5 (2/2)



`glOrtho(-4,4, -4,4, 2,20)`



`glFrustum(-1,1, -1,1, 2,20);`

Помощни функции - gluOrtho2D

```
void gluOrtho2D (  
    GLdouble left, GLdouble right,  
    GLdouble bottom, GLdouble top)
```

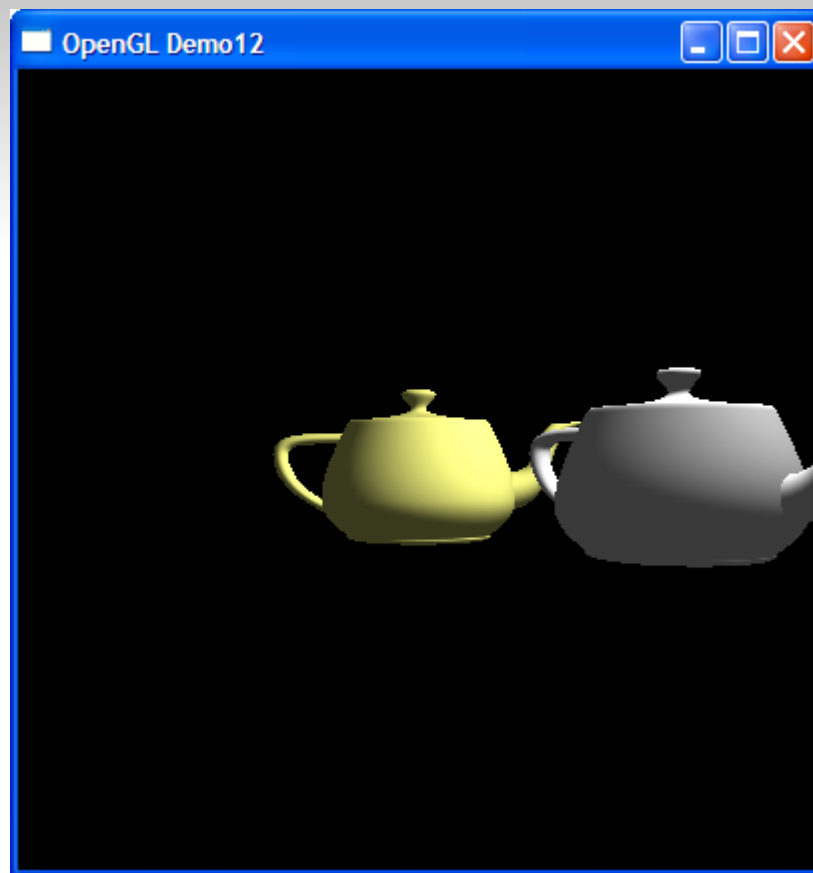
Пример:

```
gluOrtho2D (-1, 1, -1, 1);
```

Пример 6 (1/2)

```
...  
glLoadIdentity();  
glColor3d(1,1,0.5);  
glutSolidTeapot(1);  
glRotated(angle, 0,1,0);  
glTranslated(3,0,0);  
glColor3d(1,1,1);  
glutSolidTeapot(1);  
...  
  
// В resize()  
gluPerspective(45, width/height, 2, 20);
```

Пример 6 (2/2)



Помощни функции - gluUnProject

```
GLint gluUnProject(  
    GLdouble winX, GLdouble winY, GLdouble winZ,  
    const GLdouble *model,  
    const GLdouble *proj,  
    const GLint *view,  
    GLdouble* objX, GLdouble* objY, GLdouble* objZ)
```

```
GLint gluUnProject4(  
    GLdouble winX, GLdouble winY, GLdouble winZ, GLdouble clipW,  
    const GLdouble *model,  
    const GLdouble *proj,  
    const GLint *view,  
    GLdouble nearVal, GLdouble farVal,  
    GLdouble* objX, GLdouble* objY, GLdouble* objZ, GLdouble* objW)
```


Помощни функции - gluProject

```
GLint gluProject(  
    GLdouble objX, GLdouble objY, GLdouble objZ,  
    const GLdouble *model,  
    const GLdouble *proj,  
    const GLint *view,  
    GLdouble* winX, GLdouble* winY, GLdouble* winZ)
```

Пример 7 (1/2)

```
void mouse(int button, int state, int x, int y) {
    GLdouble modelMatrix[16];
    GLdouble projMatrix[16];
    GLint viewport[4];
    GLdouble objx, objy, objz;
    GLfloat z;

    if (button==GLUT_LEFT_BUTTON && state==GLUT_DOWN) {
        glGetDoublev(GL_MODELVIEW_MATRIX, modelMatrix);
        glGetDoublev(GL_PROJECTION_MATRIX, projMatrix);
        glGetIntegerv(GL_VIEWPORT, viewport);

        glReadPixels(x, y, 1, 1, GL_DEPTH_COMPONENT, GL_FLOAT, &z);

        gluUnProject(x,viewport[3]-y,z,
                    modelMatrix, projMatrix, viewport,
                    &objx, &objy, &objz);
        printf("x=%g, y=%g, z=%g\n", objx, objy, objz);
    }
}

// В main()
glutMouseFunc(mouse);
```

Пример 7 (2/2)



```
x=-0.59, y=-0.06, z=0.714627  
x=-0.36, y=-0.05, z=0.853984  
x=0.14, y=0, z=0.929592  
x=0.43, y=0.1, z=0.871285  
x=0.55, y=0.21, z=0.826688  
x=0.59, y=0.26, z=0.804585  
x=0.08, y=0.56, z=0.872948  
x=0.03, y=0.82, z=-7  
x=0.62, y=0.99, z=-7
```

OpenGL - Трансформации

Въпроси?