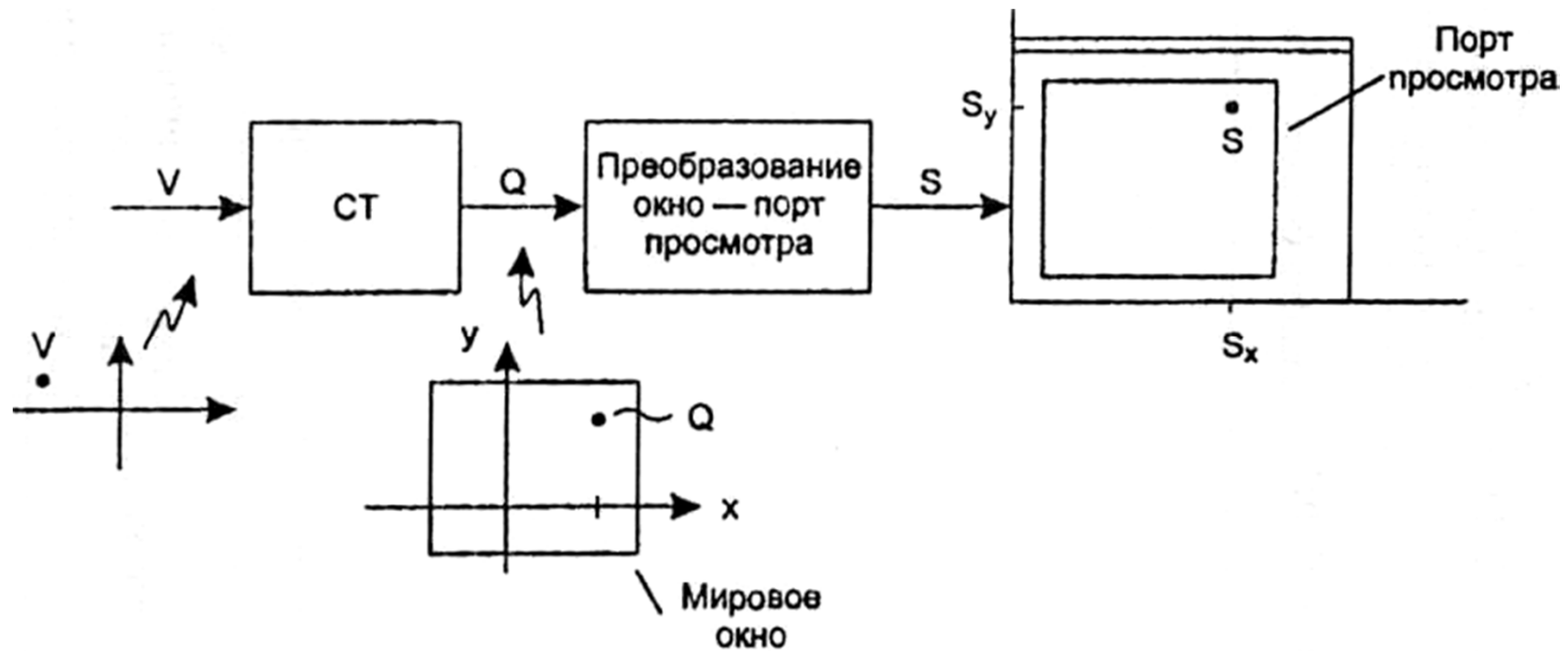


# КООРДИНАТНЫЕ ПРЕОБРАЗОВАНИЯ В OPENGGL



# ПРЕОБРАЗОВАНИЯ В OPENGGL

Функции, обычно используемые для модификации матрицы моделирования-вида

- `glScaled(sx, sy, sz)`; умножает справа текущую матрицу на матрицу, выполняющую масштабирование на величину  $sx$  по оси  $x$ , на  $sy$  по оси  $y$  и на  $sz$  по оси  $z$ . Результат помещается обратно в текущую матрицу.
- `glTranslated(dx, dy, dz)`; умножает справа текущую матрицу на матрицу, которая выполняет перемещение на  $dx$  по оси  $x$ , на  $dy$  по оси  $y$  и на  $dz$  по оси  $z$ . Результат помещается обратно в текущую матрицу.
- `glRotated (angle, ux, uy, uz)`; умножает справа текущую матрицу на матрицу, которая выполняет поворот на  $angle$  градусов вокруг оси, которая проходит через начало координат и точку  $(ux, uy, uz)$ . Результат помещается обратно в текущую матрицу.

# ПРЕОБРАЗОВАНИЯ В OPENGGL

В случае двух измерений:

- `glScaled(sx, sy, 1.0);` умножает СТ справа на матрицу, выполняющую масштабирование на величину `sx` по оси `x` и на величину `sy` по оси `y`; результат помещается обратно в СТ. По оси `z` никакого масштабирования не производится.
- `glTranslated(dx, dy, 0);` умножает СТ справа на матрицу, выполняющую перемещение на `dx` по оси `x` и на `dy` по оси `y`; результат помещается обратно в СТ. По оси `z` никакого перемещения не производится.
- `glRotated (angle, 0, 0, 1);` умножает СТ справа на матрицу, выполняющую поворот на `angle` градусов вокруг оси `z` (она обозначена `(0,0,1)`); результат помещается обратно в СТ.

# ПРЕОБРАЗОВАНИЯ В OPENGL

Основными подпрограммами для поддержки матрицы моделирования-вида являются `glRotated()`, `glScaled()` и `glTranslated()`.

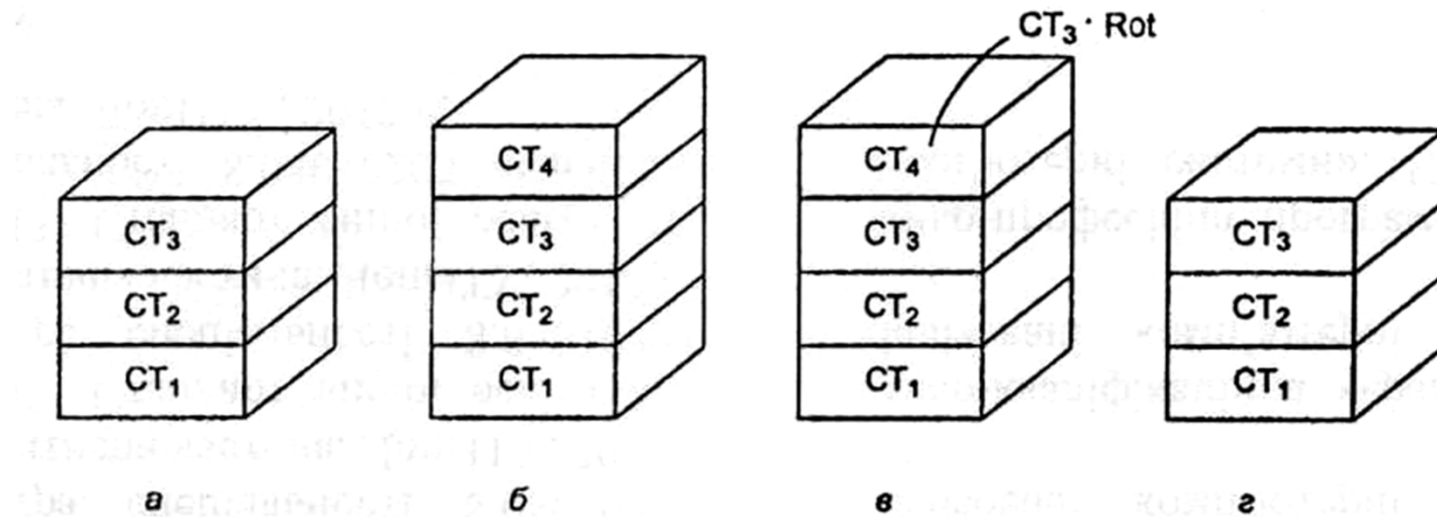
Они не устанавливают СТ прямо.

Вместо этого в каждой из них матрица моделирования-вида СТ умножается справа на заданную матрицу, например М, после чего результат помещается обратно в СТ. Таким образом, в каждой из этих подпрограмм создается матрица М, необходимая для нового преобразования, и выполняется следующая операция:

$$СТ = СТ * М$$

Порядок следования здесь имеет значение.

# ПРЕОБРАЗОВАНИЯ В OPENGL



```
glMatrixMode(GL_MODELVIEW);  
glPushMatrix();
```

...

```
glMatrixMode(GL_MODELVIEW);  
glPopMatrix();
```

# ПРЕОБРАЗОВАНИЯ В OPENGL

```
GLfloat matrix[16];           // Прекрасная матрица
                               // с точки зрения OpenGL
GLfloat matrix[4][4];         // Вариант популярный, но не сильно
                               // эффективный в OpenGL
```

$$\begin{array}{c} \text{Направление оси } x \\ \downarrow \\ x_x \\ x_y \\ x_z \\ 0 \end{array} \begin{array}{c} \text{Направление оси } y \\ \downarrow \\ y_x \\ y_y \\ y_z \\ 0 \end{array} \begin{array}{c} \text{Направление оси } z \\ \downarrow \\ z_x \\ z_y \\ z_z \\ 0 \end{array} \begin{array}{c} \text{Трансляция/положение} \\ \downarrow \\ t_x \\ t_y \\ t_z \\ 1 \end{array} \left[ \begin{array}{cccc} x_x & y_x & z_x & t_x \\ x_y & y_y & z_y & t_y \\ x_z & y_z & z_z & t_z \\ 0 & 0 & 0 & 1 \end{array} \right]$$

$$\left[ \begin{array}{cccc} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{array} \right]$$

# ПРЕОБРАЗОВАНИЯ В OPENGL

```
glLoadMatrixf(GLfloat m);  
glLoadMatrixd(GLdouble m);
```

// Загружается единичная матрица

```
GLfloat m[] = {1.0f, 0.0f, 0.0f, 0.0f, //СтолбецX  
               0.0f, 1.0f, 0.0f, 0.0f, //Столбец Y  
               0.0f, 0.0f, 1.0f, 0.0f, //СтолбецZ  
               0.0f, 0.0f, 0.0f, 1.0f}; //Трансляция
```

# ПРЕОБРАЗОВАНИЯ В OPENGL

```
glMatrixMode(GL_MODELVIEW);  
glLoadMatrixf(m);  
void glLoadTransposeMatrixf(GLfloat m);  
void glLoadTransposeMatrixd(GLdouble m);  
...  
glPushMatrix();  
glMultMatrixf(translationMatrix);  
glMultMatirxf(rotationMatrix);  
...  
glPopMatrix();
```