

NAME

glOrtho – multiply the current matrix with an orthographic matrix

C SPECIFICATION

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

PARAMETERS

left, right

Specify the coordinates for the left and right vertical clipping planes.

bottom, top

Specify the coordinates for the bottom and top horizontal clipping planes.

zNear, zFar

Specify the distances to the nearer and farther depth clipping planes. These values are negative if the plane is to be behind the viewer.

DESCRIPTION

glOrtho describes a transformation that produces a parallel projection. The current matrix (see **glMatrixMode**) is multiplied by this matrix and the result replaces the current matrix, as if **glMultMatrix** were called with the following matrix as its argument:

```
left ( matrix {
  ccol { 2 over {"right" - "left"} } above 0 above 0 above 0 }
  ccol { 0 above {2 over {"top" - "bottom"}} above 0 above 0 }
  ccol { 0 above 0 above {-2 over {"zFar" - "zNear"}} above 0 }
  ccol { {t sub x}~ above {t sub y}~ above {t sub z}~ above 1~ } } right )
```

where

$$t \text{ sub } x \sim = -\{ \{ "right" + "left" \} \text{ over } \{ "right" - "left" \} \}$$

$$t \text{ sub } y \sim = -\{ \{ "top" + "bottom" \} \text{ over } \{ "top" - "bottom" \} \}$$

$$t \text{ sub } z \sim = -\{ \{ "zFar" + "zNear" \} \text{ over } \{ "zFar" - "zNear" \} \}$$

Typically, the matrix mode is **GL_PROJECTION**, and (*left, bottom, -zNear*) and (*right, top, -zNear*) specify the points on the near clipping plane that are mapped to the lower left and upper right corners of the window, respectively, assuming that the eye is located at (0, 0, 0). *-zFar* specifies the location of the far clipping plane. Both *zNear* and *zFar* can be either positive or negative.

Use **glPushMatrix** and **glPopMatrix** to save and restore the current matrix stack.

ERRORS

GL_INVALID_OPERATION is generated if **glOrtho** is executed between the execution of **glBegin** and the corresponding execution of **glEnd**.

ASSOCIATED GETS

glGet with argument **GL_MATRIX_MODE**

glGet with argument **GL_MODELVIEW_MATRIX**

glGet with argument **GL_PROJECTION_MATRIX**

glGet with argument **GL_TEXTURE_MATRIX**

SEE ALSO

glFrustum, glMatrixMode, glMultMatrix, glPushMatrix, glViewport