

NAME

glRotated, **glRotatef** – multiply the current matrix by a rotation matrix

C SPECIFICATION

```
void glRotated( GLdouble angle,
                GLdouble x,
                GLdouble y,
                GLdouble z )
void glRotatef( GLfloat angle,
                GLfloat x,
                GLfloat y,
                GLfloat z )
```

delim \$\$

PARAMETERS

angle Specifies the angle of rotation, in degrees.

x, *y*, *z* Specify the *x*, *y*, and *z* coordinates of a vector, respectively.

DESCRIPTION

glRotate produces a rotation of *angle* degrees around the vector (x, y, z) . The current matrix (see **glMatrixMode**) is multiplied by a rotation matrix with the product replacing the current matrix, as if **glMultMatrix** were called with the following matrix as its argument:

```
left ( ~ down 20 matrix {
ccol { "x" "x" (1 - c)+ c above "y" "x" (1 - c)+ "z" s above "x" "z" (1 - c)-"y" s above ~0 }
ccol { "x" "y" (1 - c)-"z" s above "y" "y" (1 - c)+ c above "y" "z" (1 - c)+ "x" s above ~0 }
ccol { "x" "z" (1 - c)+ "y" s above "y" "z" (1 - c)- "x" s above "z" "z" (1 - c) + c above ~0 }
ccol { ~0 above ~0 above ~0 above ~1 } } ~ right )
```

Where $c = \cos(\text{angle})$, $s = \sin(\text{angle})$, and $\|(\text{"x"}, \text{"y"}, \text{"z"})\| = 1$ (if not, the GL will normalize this vector).

If the matrix mode is either **GL_MODELVIEW** or **GL_PROJECTION**, all objects drawn after **glRotate** is called are rotated. Use **glPushMatrix** and **glPopMatrix** to save and restore the unrotated coordinate system.

NOTES

This rotation follows the right-hand rule, so if the vector (x, y, z) points toward the user, the rotation will be counterclockwise.

ERRORS

GL_INVALID_OPERATION is generated if **glRotate** 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

glMatrixMode, **glMultMatrix**, **glPushMatrix**, **glScale**, **glTranslate**