NAME

gluLookAt – define a viewing transformation

C SPECIFICATION

void **gluLookAt**(GLdouble *eyeX*,
GLdouble *eyeY*,
GLdouble *eyeZ*,
GLdouble *centerX*,
GLdouble *centerY*,
GLdouble *centerZ*,
GLdouble *upX*,
GLdouble *upY*,
GLdouble *upZ*)

delim $$

PARAMETERS

eyeX, *eyeY*, *eyeZ*
Specifies the position of the eye point.

centerX, *centerY*, *centerZ*
Specifies the position of the reference point.

upX, *upY*, *upZ* Specifies the direction of the *up* vector.

DESCRIPTION

gluLookAt creates a viewing matrix derived from an eye point, a reference point indicating the center of the scene, and an *UP* vector.

The matrix maps the reference point to the negative *z* axis and the eye point to the origin. When a typical projection matrix is used, the center of the scene therefore maps to the center of the viewport. Similarly, the direction described by the *UP* vector projected onto the viewing plane is mapped to the positive *y* axis so that it points upward in the viewport. The *UP* vector must not be parallel to the line of sight from the eye point to the reference point.

Let

F ~=~ left (down 20 { ~ matrix {
ccol {"centerX" above "centerY" above "centerZ"}
ccol { ~-~ above ~-~ above ~-~}
ccol {"eyeX" above "eyeY" above "eyeZ"} } } ~~ right)

Let *UP* be the vector $("upX", "upY", "upZ")$.

Then normalize as follows: f ~=~ F over {|| F ||}

UP sup prime ~=~ UP over {|| UP ||}

Finally, let $s ~=~ f ~times~ UP sup prime$, and $u ~=~ s ~times~ f$.

M is then constructed as follows: M ~=~ left (matrix {
ccol { ~s[0] above ~u[0] above -f[0] above 0 }
ccol { ~s[1] above ~u[1] above -f[1] above 0 }
ccol { ~s[2] above ~u[2] above -f[2] above 0 }
ccol { 0 above 0 above 0 above 1 } } ~~right)

and **gluLookAt** is equivalent to glMultMatrixf(M); glTranslated (-eyex, -eyey, -eyez);

SEE ALSO

glFrustum, **gluPerspective**