Name

glTexGen -- control the generation of texture coordinates

C Specification

```
void glTexGeni(coord, pname, param);
GLenum coord;
GLenum pname;
GLint param;

void glTexGenf(coord, pname, param);
GLenum coord;
GLenum pname;
GLfloat param;

void glTexGend(coord, pname, param);
GLenum coord;
GLenum pname;
GLdouble param;
```

PARAMETERS

coord *coord*
Specifies a texture coordinate. Must be one of GL_S, GL_T, GL_R, or GL_Q.

pname *pname*
Specifies the symbolic name of the texture-coordinate generation function. Must be GL_TEXTURE_GEN_MODE.

param *param*
Specifies a single-valued texture generation parameter, one of GL_OBJECT_LINEAR, GL_EYE_LINEAR, GL_SPHERE_MAP, GL_NORMAL_MAP, or GL_REFLECTION_MAP.

C Specification

```
void glTexGeniv(coord, pname, params);
GLenum coord;
GLenum pname;
const GLint * params;

void glTexGenfv(coord, pname, params);
GLenum coord;
GLenum pname;
const GLfloat * params;

void glTexGendv(coord, pname, params);
GLenum coord;
GLenum pname;
const GLdouble * params;
```

PARAMETERS

coord *coord*

Specifies a texture coordinate. Must be one of GL_S, GL_T, GL_R, or GL_Q.

pname *pname*
Specifies the symbolic name of the texture-coordinate generation function or function parameters. Must be GL_TEXTURE_GEN_MODE, GL_OBJECT_PLANE, or GL_EYE_PLANE.

params *params*
Specifies a pointer to an array of texture generation parameters. If *pname* is GL_TEXTURE_GEN_MODE, then the array must contain a single symbolic constant, one of GL_OBJECT_LINEAR, GL_EYE_LINEAR, GL_SPHERE_MAP, GL_NORMAL_MAP, or GL_REFLECTION_MAP. Otherwise, *params* holds the coefficients for the texture-coordinate generation function specified by *pname*.

DESCRIPTION

glTexGen selects a texture-coordinate generation function or supplies coefficients for one of the functions. *coord* names one of the (*s*, *t*, *r*, *q*) texture coordinates; it must be one of the symbols GL_S, GL_T, GL_R, or GL_Q. *pname* must be one of three symbolic constants: GL_TEXTURE_GEN_MODE, GL_OBJECT_PLANE, or GL_EYE_PLANE. If *pname* is GL_TEXTURE_GEN_MODE, then *params* chooses a mode, one of GL_OBJECT_LINEAR, GL_EYE_LINEAR, GL_SPHERE_MAP, GL_NORMAL_MAP, or GL_REFLECTION_MAP. If *pname* is either GL_OBJECT_PLANE or GL_EYE_PLANE, *params* contains coefficients for the corresponding texture generation function.

If the texture generation function is GL_OBJECT_LINEAR, the function

is used, where is the value computed for the coordinate named in *coord*, , , , and are the four values supplied in *params*, and , , , and are the object coordinates of the vertex. This function can be used, for example, to texture-map terrain using sea level as a reference plane (defined by , , , and). The altitude of a terrain vertex is computed by the GL_OBJECT_LINEAR coordinate generation function as its distance from sea level; that altitude can then be used to index the texture image to map white snow onto peaks and green grass onto foothills.

If the texture generation function is GL_EYE_LINEAR, the function

is used, where

and , , , and are the eye coordinates of the vertex, , , , and are the values supplied in *params*, and is the modelview matrix when glTexGen is invoked. If is poorly conditioned or singular, texture coordinates generated by the resulting function may be inaccurate or undefined.

Note that the values in *params* define a reference plane in eye coordinates. The modelview matrix that is applied to them may not be the same one in effect when the polygon vertices are transformed. This function establishes a field of texture coordinates that can produce dynamic contour lines on moving objects.

If *pname* is GL_SPHERE_MAP and *coord* is either GL_S or GL_T, and texture coordinates are generated as follows. Let *u* be the unit vector pointing from the origin to the polygon vertex (in eye coordinates). Let *n* sup prime be the current normal, after transformation to eye coordinates. Let

be the reflection vector such that

Finally, let . Then the values assigned to the and texture coordinates are

To enable or disable a texture-coordinate generation function, call glEnable or glDisable with one of the symbolic texture-coordinate names (GL_TEXTURE_GEN_S, GL_TEXTURE_GEN_T, GL_TEXTURE_GEN_R, or GL_TEXTURE_GEN_Q) as the argument. When enabled, the specified texture coordinate is computed according to the generating function associated with that coordinate. When disabled, subsequent vertices take the specified texture coordinate from the current set of texture coordinates. Initially, all texture generation functions are set to GL_EYE_LINEAR and are disabled. Both plane equations are (1, 0, 0, 0), both plane equations are (0, 1, 0, 0), and all and plane equations are (0, 0, 0, 0).

When the ARB_multitexture extension is supported, glTexGen set the texture generation parameters for the currently active texture unit, selected with glActiveTexture.

ERRORS

GL_INVALID_ENUM is generated when *coord* or *pname* is not an accepted defined value, or when *pname* is GL_TEXTURE_GEN_MODE and *params* is not an accepted defined value.

GL_INVALID_ENUM is generated when *pname* is GL_TEXTURE_GEN_MODE, *params* is GL_SPHERE_MAP, and *coord* is either GL_R or GL_Q.

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

ASSOCIATED GETS

glGetTexGen

glIsEnabled with argument GL_TEXTURE_GEN_S

glIsEnabled with argument GL_TEXTURE_GEN_T

glIsEnabled with argument GL_TEXTURE_GEN_R

glIsEnabled with argument GL_TEXTURE_GEN_Q

SEE ALSO

glActiveTexture, glCopyPixels, glCopyTexImage2D, glCopyTexSubImage1D, glCopyTexSubImage2D, glCopyTexSubImage3D, glTexEnv, glTexImage1D, glTexImage2D, glTexImage3D, glTexParameter, glTexSubImage1D, glTexSubImage2D, glTexSubImage3D

COPYRIGHT