

NAME

glEvalCoord1d, **glEvalCoord1f**, **glEvalCoord2d**, **glEvalCoord2f**, **glEvalCoord1dv**, **glEvalCoord1fv**, **glEvalCoord2dv**, **glEvalCoord2fv** – evaluate enabled one- and two-dimensional maps

delim \$\$

C SPECIFICATION

```
void glEvalCoord1d( GLdouble u )
void glEvalCoord1f( GLfloat u )
void glEvalCoord2d( GLdouble u,
                    GLdouble v )
void glEvalCoord2f( GLfloat u,
                    GLfloat v )
```

PARAMETERS

- u* Specifies a value that is the domain coordinate \$u\$ to the basis function defined in a previous **glMap1** or **glMap2** command.
- v* Specifies a value that is the domain coordinate \$v\$ to the basis function defined in a previous **glMap2** command. This argument is not present in a **glEvalCoord1** command.

C SPECIFICATION

```
void glEvalCoord1dv( const GLdouble *u )
void glEvalCoord1fv( const GLfloat *u )
void glEvalCoord2dv( const GLdouble *u )
void glEvalCoord2fv( const GLfloat *u )
```

PARAMETERS

- u* Specifies a pointer to an array containing either one or two domain coordinates. The first coordinate is \$u\$. The second coordinate is \$v\$, which is present only in **glEvalCoord2** versions.

DESCRIPTION

glEvalCoord1 evaluates enabled one-dimensional maps at argument *u*. **glEvalCoord2** does the same for two-dimensional maps using two domain values, *u* and *v*. To define a map, call **glMap1** and **glMap2**; to enable and disable it, call **glEnable** and **glDisable**.

When one of the **glEvalCoord** commands is issued, all currently enabled maps of the indicated dimension are evaluated. Then, for each enabled map, it is as if the corresponding GL command had been issued with the computed value. That is, if **GL_MAP1_INDEX** or **GL_MAP2_INDEX** is enabled, a **glIndex** command is simulated. If **GL_MAP1_COLOR_4** or **GL_MAP2_COLOR_4** is enabled, a **glColor** command is simulated. If **GL_MAP1_NORMAL** or **GL_MAP2_NORMAL** is enabled, a normal vector is produced, and if any of **GL_MAP1_TEXTURE_COORD_1**, **GL_MAP1_TEXTURE_COORD_2**, **GL_MAP1_TEXTURE_COORD_3**, **GL_MAP1_TEXTURE_COORD_4**, **GL_MAP2_TEXTURE_COORD_1**, **GL_MAP2_TEXTURE_COORD_2**, **GL_MAP2_TEXTURE_COORD_3**, or **GL_MAP2_TEXTURE_COORD_4** is enabled, then an appropriate **glTexCoord** command is simulated.

For color, color index, normal, and texture coordinates the GL uses evaluated values instead of current values for those evaluations that are enabled, and current values otherwise. However, the evaluated values do not update the current values. Thus, if **glVertex** commands are interspersed with **glEvalCoord** commands, the color, normal, and texture coordinates associated with the **glVertex** commands are not affected by the values generated by the **glEvalCoord** commands, but only by the most recent **glColor**, **glIndex**, **glNormal**, and **glTexCoord** commands.

No commands are issued for maps that are not enabled. If more than one texture evaluation is enabled for a particular dimension (for example, **GL_MAP2_TEXTURE_COORD_1** and

GL_MAP2_TEXTURE_COORD_2), then only the evaluation of the map that produces the larger number of coordinates (in this case, **GL_MAP2_TEXTURE_COORD_2**) is carried out. **GL_MAP1_VERTEX_4** overrides **GL_MAP1_VERTEX_3**, and **GL_MAP2_VERTEX_4** overrides **GL_MAP2_VERTEX_3**, in the same manner. If neither a three- nor a four-component vertex map is enabled for the specified dimension, the **glEvalCoord** command is ignored.

If you have enabled automatic normal generation, by calling **glEnable** with argument **GL_AUTO_NORMAL**, **glEvalCoord2** generates surface normals analytically, regardless of the contents or enabling of the **GL_MAP2_NORMAL** map. Let

$$\begin{matrix} P_p & P_p \\ m = & \text{-- } X \text{ --} \\ P_u & P_v \end{matrix}$$

Then the generated normal \$ n \$ is

$$n = \frac{m \times \tilde{m}}{\|m \times \tilde{m}\|}$$

If automatic normal generation is disabled, the corresponding normal map **GL_MAP2_NORMAL**, if enabled, is used to produce a normal. If neither automatic normal generation nor a normal map is enabled, no normal is generated for **glEvalCoord2** commands.

ASSOCIATED GETS

glIsEnabled with argument **GL_MAP1_VERTEX_3**
glIsEnabled with argument **GL_MAP1_VERTEX_4**
glIsEnabled with argument **GL_MAP1_INDEX**
glIsEnabled with argument **GL_MAP1_COLOR_4**
glIsEnabled with argument **GL_MAP1_NORMAL**
glIsEnabled with argument **GL_MAP1_TEXTURE_COORD_1**
glIsEnabled with argument **GL_MAP1_TEXTURE_COORD_2**
glIsEnabled with argument **GL_MAP1_TEXTURE_COORD_3**
glIsEnabled with argument **GL_MAP1_TEXTURE_COORD_4**
glIsEnabled with argument **GL_MAP2_VERTEX_3**
glIsEnabled with argument **GL_MAP2_VERTEX_4**
glIsEnabled with argument **GL_MAP2_INDEX**
glIsEnabled with argument **GL_MAP2_COLOR_4**
glIsEnabled with argument **GL_MAP2_NORMAL**
glIsEnabled with argument **GL_MAP2_TEXTURE_COORD_1**
glIsEnabled with argument **GL_MAP2_TEXTURE_COORD_2**
glIsEnabled with argument **GL_MAP2_TEXTURE_COORD_3**
glIsEnabled with argument **GL_MAP2_TEXTURE_COORD_4**
glIsEnabled with argument **GL_AUTO_NORMAL**
glGetMap

SEE ALSO

glBegin, **glColor**, **glEnable**, **glEvalMesh**, **glEvalPoint**, **glIndex**, **glMap1**, **glMap2**, **glMapGrid**, **glNormal**, **glTexCoord**, **glVertex**