

Renderizando con texturas

Veamos ahora cómo podemos dibujar polígonos con texturas aplicadas. Supongamos que queremos dibujar un simple cuadrado, con la textura que hemos cargado anteriormente.

Si lo dibujamos sin textura sería:

```
glBegin (GL_QUADS);
glVertex3i (-100, -100, 5);
glVertex3i (-100, 100, 5);
glVertex3i (100, 100, 5);
glVertex3i (100, -100, 5);
glEnd ();
```

Veamos ahora como lo haríamos aplicando una textura:

```
glEnable(GL_TEXTURE_2D);
glBindTexture(GL_TEXTURE_2D, idTextura);
glBegin (GL_QUADS);
    glTexCoord2f(0.0f, 0.0f);    glVertex3i (-100, -100, 5);
    glTexCoord2f(1.0f, 0.0f);    glVertex3i (-100, 100, 5);
    glTexCoord2f(1.0f, 1.0f);    glVertex3i (100, 100, 5);
    glTexCoord2f(0.0f, 1.0f);    glVertex3i (100, -100, 5);
glEnd ();
glDisable(GL_TEXTURE_2D);
```

Analizemos el código paso por paso:

```
glEnable(GL_TEXTURE_2D);
```

La función `glEnable` de OpenGL nos permite activar y desactivar propiedades que se aplican a la hora de renderizar. En este caso, le estamos diciendo que active la texturización.

```
glBindTexture(GL_TEXTURE_2D, idTextura);
```

La función `glBindTexture` se encarga de activar la textura que deseemos, referenciada por el identificador que habíamos guardado previamente en `idTextura`.

```
glBegin (GL_QUADS);
glTexCoord2f(0.0f, 0.0f);
glVertex3i (-100, -100, 5);
```

Aquí vemos algo que no estaba antes, nos referimos a `glTexCoord2f()`. Esta función nos introduce un concepto nuevo: las coordenadas de textura, y nos referiremos a ellas generalmente como 's' y 't', siendo 's' el eje horizontal y 't' el vertical. Se usan para referirnos a una posición de la textura.

Generalmente se mueven en el intervalo [0, 1]. La coordenada (0, 0) se refiere a la esquina inferior izquierda, y la (1, 1) a la superior derecha. Por lo tanto, si nos fijamos en el código, a cada vértice le asignamos una coordenada de textura, siempre en orden contrario a las agujas del reloj, sino la textura se vería al revés.

Hemos comentado que las coordenadas de textura se mueven en el intervalo [0, 1], pero OpenGL permite otros valores. Podríamos considerar que la textura se repite infinitamente en todas las direcciones, de manera que la coordenada (1, 0) sería el mismo texel que (2, 0), (3, 0), etc. No obstante, si dibujamos un cuadrado y le asignamos a un extremo (2, 2) en lugar de (1, 1), dibujara la textura repetida 2 veces en cada dirección, en total, 4 veces. Podemos jugar y asignar valores de (-1, -1) a (1, 1), etc.

Una vez hayamos dibujado la geometría que queramos, es importante desactivar la texturización, si no lo

hacemos, si mas adelante queremos dibujar algún objeto sin texturizar, OpenGL intentará hacerlo, aunque no le pasemos coordenadas de textura, produciendo efectos extraños. Así pues:

```
glDisable(GL_TEXTURE_2D);
```