

Textures amb OpenGL

© Professors de VA

Grup MOVING – Dep. LSI – UPC

Ús de textures

- Tres etapes:
 - **Creació** de la textura:
 - *Creació*: glGenTexture, glBindTexture, glTexImage
 - *Definició paràmetres*: glTexParameter
 - **Dibuix** de les primitives texturades
 - *Activació*: glEnable i glBindTexture
 - *Definició funció texturació*: glTexEnvf
 - *Generació coordenades*: glTexCoord o automàtiques
 - **Destrucció** textures: glDeleteTextures

Ús de textures

// 1. Activar el texture mapping desitjat

// Només pot estar activat un mode: GL_TEXTURE_1D, 2D o 3D

```
glEnable(GL_TEXTURE_2D);
```

// 2. Activar el texture object corresponent

```
glBindTexture(GL_TEXTURE_2D, id);
```

// 3. Establir la funció de texturació

```
glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE);
```

// 4. Dibuixar la primitiva

```
glBegin(GL_POLYGON);
```

```
glTexCoord2d(0, 0);
```

```
glVertex3d(...);
```

```
...
```

// o utilitzant coordenades automàtiques

Creació de l'objecte textura

- Generar un nou nom:
 - void **glGenTextures**(1, &*texName*);
 - Crea una textura (1) i emmagatzema el seu identificador a *texName*
- Activar la textura
 - void **glBindTexture**(GL_TEXTURE_2D, *texName*);
 - Les següents operacions de textures actuaran sobre *texName*.

Creació de l'objecte textura

- Introducció de les dades:
 - void **glTexImage2D**(GLenum *objective*, GLint *level*, GLint *internalFormat*, GLsizei *width*, GLsizei *height*, GLint *border*, GLenum *format*, GLenum *type*, GLvoid* *pixels*);
 - *objective*: GL_TEXTURE_2D
 - *level*: 0 (nivells de mip mapping)
 - *internalFormat* i *format*: GL_RGB o GL_RGBA
 - *width* i *height*: de la forma $2^m + 2^b$ (mín 64x64)
 - *border*: 0 o 1
 - *type*: de les dades que passen a *pixels* (GL_BYTE, GL_FLOAT...)
 - *pixels*: Array de bytes amb valors del tipus *tipus*

Creació de l'objecte textura

- Altres formes de posar les dades. A partir de la informació generada:
 - void **glCopyTexImage2D**(GLenum target, GLint level, GLenum internalFormat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border);
 - Defineix la textura a partir d'una regió rectangular del GL_READ_BUFFER actiu (com el *glCopyPixels* però els *pixels* van a memòria de textura en comptes del *framebuffer*).
 - void **glCopyTexSubImage2D**(GLenum *target*, GLint *level*, GLint *xoffset*, GLint *yoffset*, GLint x, GLint y, GLsizei *width*, GLsizei *height*);
 - Substitueix una regió rectangular d'una textura ja definida per una regió rectangular.

Funcions de textura

- Definir el comportament en filtrat:
 - void **glTexParameterf**(GL_TEXTURE_2D, *filtre*, *filtrat*);
 - *filtre*: ampliació (GL_TEXTURE_MAG_FILTER) o reducció (GL_TEXTURE_MIN_FILTER)
 - *filtrat*: agafar el més proper (GL_NEAREST) o una interpolació (GL_LINEAR)
 - Si no es defineixen els filtres pot no veure's res!!!

Dibuixat escena

- Definir coordenades:
 - Primer es posen les coordenades de textura del vèrtex:
 - **glCoord2f** (coordTexX, coordTexY);
 - Després es projecta el vèrtex:
 - **glVertex3f** (coordX, coordY, coordZ);

Dibuixat escena

- Comportament més enllà de [0.0, 1.0]:
 - void **glTexParameter**i(GL_TEXTURE_2D, *param*, *tipus*);
 - *param*: s (GL_TEXTURE_WRAP_S) o t (GL_TEXTURE_WRAP_T)
 - *tipus*: repetir (GL_REPEAT) o tallar (GL_CLAMP)