

Tema 10: Interacción

J. Ribelles

SIE020: Síntesis de Imagen y Animación
Institute of New Imaging Technologies, Universitat Jaume I

Contenido

- 1 Introducción
- 2 Eventos
- 3 Menús
- 4 Selección

Introducción

Introducción

- La captura y el manejo de eventos es esencial para crear aplicaciones interactivas.
- OpenGL no proporciona ninguna herramienta que nos permita capturar eventos.
- Pero la librería FreeGLUT si.
- Y también crear menús desplegados.
- Con OpenGL sí que podemos seleccionar objetos del mundo 3D.

Eventos

Descripción

- Una de las finalidades de la librería FreeGLUT es precisamente dar ese soporte.
- Funciones relacionadas con los eventos de teclado:
 - `void glutKeyboardFunc(void(*f)(unsigned char tecla,int x, int y));`
 - `void glutSpecialFunc(void (*f)(int tecla, int x, int y));`
- Funciones relacionadas con los eventos del ratón:
 - `void glutMouseFunc(void (*f)(int boton, int estado, int x, int y));`
 - `void glutMotionFunc(void (*f)(int x, int y));`
- Funciones relacionadas con los eventos de ventana:
 - `void glutReshapeFunc(void (*f)(int ancho, int alto));`
- Además, con la orden `glutPostRedisplay()` el programador puede generar un evento para solicitar el redibujado de la ventana.

Menús

```
// Crea un submenu  
idSubmenu = glutCreateMenu(menu);  
  
// Las opciones del submenu  
glutAddMenuEntry("Alzado", 2);  
glutAddMenuEntry("Planta", 3);  
glutAddMenuEntry("Perfil", 4);  
  
// Crea el menu principal  
idMenu = glutCreateMenu(menu);  
  
// Crea las entradas  
glutAddMenuEntry("Perspectiva", 1);  
glutAddSubMenu ("Paralela", idSubmenu);  
glutAddMenuEntry("Salir", 0);  
  
// Asocia el menu al boton derecho  
glutAttachMenu(GLUT_RIGHT_BUTTON);
```



Selección

Descripción

- ¿Cómo señalar objetos de la escena y que por tanto la aplicación pueda saber qué objeto se ha señalado?
- Una forma de dar soporte a la selección de objetos con OpenGL es crear un nuevo objeto *framebuffer*.
- Consiste en dibujar la misma escena pero en el nuevo objeto *framebuffer* y con una salvedad, los objetos seleccionables se pintarán con colores planos y diferentes entre sí.
- Después, utilizando las coordenadas de ventana proporcionadas por la FreeGLUT tras el click realizado por el usuario, se leerá el color de dicho píxel y así sabremos de qué objeto se trata.
- El nuevo objeto *framebuffer* debe constar de buffer de color y de profundidad, y tendrá la misma dimensión que el *framebuffer* de la ventana de la aplicación.

Creación de un objeto framebuffer

```
enum { Color , Depth , NumRenderbuffers };
GLuint framebuffer , renderbuffer[NumRenderbuffers];

// recibe como parametros el ancho y el alto del framebuffer
void initFramebuffer (int ancho , int alto) {
    // se crean dos buffers
    glGenRenderbuffers(NumRenderbuffers , renderbuffer);

    // uno almacenara el color
    glBindRenderbuffer (GL_RENDERBUFFER , renderbuffer[Color]);
    glRenderbufferStorage(GL_RENDERBUFFER , GL_RGBA , ancho , alto);

    // este otro almacenara la profundidad
    glBindRenderbuffer (GL_RENDERBUFFER , renderbuffer[Depth]);
    glRenderbufferStorage(GL_RENDERBUFFER , GL_DEPTH_COMPONENT24 , ancho , alto);

    // ahora creamos el framebuffer
    glGenFramebuffers(1 , &framebuffer);
    glBindFramebuffer(GL_DRAW_FRAMEBUFFER , framebuffer);

    // y le adjudicamos los dos buffers creados previamente
    glFramebufferRenderbuffer(GL_DRAW_FRAMEBUFFER , GL_COLOR_ATTACHMENT0 ,
                             GL_RENDERBUFFER , renderbuffer[Color]);
    glFramebufferRenderbuffer(GL_DRAW_FRAMEBUFFER , GL_DEPTH_ATTACHMENT ,
                             GL_RENDERBUFFER , renderbuffer[Depth]);

    glEnable(GL_DEPTH_TEST);

    glBindFramebuffer(GL_DRAW_FRAMEBUFFER , 0); // de momento lo desactivamos
}
```

Dibujado en el objeto framebuffer

```
// recibe como parametros las coordenadas del pixel  
void renderToFramebuffer (int x, int y)  
{  
    // activa el framebuffer para dibujar  
    glBindFramebuffer(GL_DRAW_FRAMEBUFFER, framebuffer);  
  
    // dibuja la escena utilizando colores planos  
    ...  
  
    // recupera el color del pixel seleccionado  
    // dependiendo del color leido sabras que objeto  
    // se ha seleccionado  
    GLfloat pixel[4];  
    glBindFramebuffer(GL_READ_FRAMEBUFFER, framebuffer);  
    glReadPixels(x, alto-y, 1, 1, GL_RGBA, GL_FLOAT, pixel);  
  
    // Desactiva de nuevo el framebuffer  
    glBindFramebuffer(GL_DRAW_FRAMEBUFFER, 0);  
}
```