

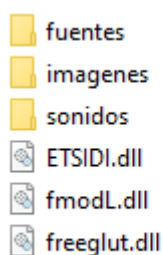
Embelluyendo el Juego con texturas y sprites

Estas instrucciones se incluyen para explicar con detalle como poder integrar la librería ETSIDlib en cualquiera de los temas de las prácticas a partir del tema 4.

Durante los capítulos anteriores se ha mantenido desacoplada la parte de dibujo de la parte lógica en todos los componentes. Esto nos permitirá ahora con muy poco esfuerzo cambiar completamente el aspecto del juego gracias a la inclusión de texturas. Para ello, en el código suministrado, se ha incluido la librería ETSIDI, creada para facilitar el uso de texturas, fuentes y sonidos.

En el fichero zip se incluyen estas instrucciones y dos directorios. El contenido de ambos directorios deberá copiarse en los correspondientes directorios bin y lib del proyecto.

Si nos fijamos, en primer lugar, se ha incluido dentro del directorio bin, los directorios que incluyen una serie de ficheros auxiliares (imágenes y sonidos), así como la librería ETSIDI, y fmodL. Esta última es la encargada de gestionar los sonidos por medio de la primera que simplifica su uso:



En segundo lugar se ha incluido la librería (.lib) y el fichero de cabecera (.h) que permite el enlazado con este código y que se incluye en el directorio de librerías.

Para que la librería funcione deben estar modificadas las opciones del proyecto que informan de que las librerías auxiliares hay que buscarlas en lib, y las de ejecución en bin. Además, si queremos usar los ficheros desde la ejecución en Visual Studio que tengan su raíz en el directorio bin es necesario incluirlo en las propiedades.

Por tanto, en las opciones del proyecto debemos asegurarnos de que:

- el Directorio de salida del apartado “General” es **\$(SolutionDir)bin**
- el Directorio de trabajo del apartado “Depuración” es **\$(ProjectDir)\bin**
- los Directorios de inclusión adicionales en “C/C++” contiene **lib**
- Los directorios de bibliotecas adicionales en “Vinculador” contiene **lib**

Durante el desarrollo de las prácticas se incluyeron estas modificaciones para poder usar las librerías freeglut, sin embargo no se modificó el Directorio de “Depuracion” el cual es necesario para poder usar los ficheros ubicados en los directorios de bin. Por ello habitualmente, esto es lo único necesario que habrá que modificar para incluir ETSIDlib en el juego desarrollado en la práctica.

MODIFICACIONES ESPECÍFICAS PARA VER EL FONDO Y LA ANIMACIÓN DEL HOMBRE EN EL JUEGO PANG

A continuación, se indican, a modo de introducción, como se podría modificar el dibujo de la caja y del hombre en el juego Pang para hacer uso del fondo y la secuencia de movimiento descrito por las imágenes incluidas en este zip. En los ficheros afectados se han marcado en rojo la parte del código modificada.

Modificaciones en Caja.cpp

```
#include "Caja.h"
#include "ETSIDI.h"

...

void Caja::dibuja()
{
    suelo.dibuja();
    techo.dibuja();
    pared_izq.dibuja();
    pared_dcha.dibuja();
    //dibujo del fondo
    glEnable(GL_TEXTURE_2D);
    glBindTexture(GL_TEXTURE_2D, ETSIDI::getTexture( "imagenes/fondo.png" ).id);
    glDisable(GL_LIGHTING);
    glBegin(GL_POLYGON);
        glColor3f(1,1,1);
        glTexCoord2d(0,1);          glVertex2f(-10,0);
        glTexCoord2d(1,1);          glVertex2f(10,0);
        glTexCoord2d(1,0);          glVertex2f(10,15);
        glTexCoord2d(0,0);          glVertex2f(-10,15);
    glEnd();
    glEnable(GL_LIGHTING);
    glDisable(GL_TEXTURE_2D);
}
```

Modificaciones en Hombre.h

```
#pragma once
#include "Vector2D.h"
#include "ETSIDI.h"
using ETSIDI::SpriteSequence;
class Hombre
{
    SpriteSequence sprite{ "imagenes/pangPlayer.png", 5 };
    float altura;
    Vector2D posicion;
    Vector2D velocidad;
    Vector2D aceleracion;
public:
    void mueve(float t);
    void dibuja();
    void setVel(float vx, float vy);
    Hombre();
    friend class Interaccion;
};
```

Modificaciones en Hombre.cpp

```
#include "freeglut.h"
#include "Hombre.h"

Hombre::Hombre()
{
    sprite.setCenter(1, 0);
    sprite.setSize(2, 2);
    altura = 1.8f;
}

void Hombre::dibuja()
{
    glPushMatrix();
    glTranslatef(posicion.x, posicion.y, 1);
    glColor3f(1.0f, 0.0f, 0.0f);
    //glutSolidSphere(altura, 20, 20);

    //gestion de direccion y animacion
    if (velocidad.x > 0.01) sprite.flip(false, false);
    if (velocidad.x < -0.01) sprite.flip(true, false);
    if ((velocidad.x < 0.01) && (velocidad.x > -0.01))
        sprite.setState(0);
    else if (sprite.getState() == 0)
        sprite.setState(1, false);
    sprite.draw();
    glPopMatrix();
}

void Hombre::mueve(float t)
{
    posicion = posicion + velocidad * t + aceleracion * (0.5f * t * t);
    velocidad = velocidad + aceleracion * t;
    sprite.loop();
}
```