



Fundamentos de los Videojuegos

Grado en Ingeniería Multimedia. Curso 2017/18.

Profesor: Fidel Aznar (fidel@dccia.ua.es). Departamento DCCIA

Práctica Individual

El objetivo de esta práctica es doble, por una parte se centra en la instalación de los recursos necesarios para elaborar la práctica y el proyecto de la asignatura. Por otra parte introduce la librería SFML y el desarrollo de un pequeño juego.

Este juego puede mejorarse para subir hasta 1,5 puntos de la nota de la asignatura.

Parte 1. Instalación de las librerías

Se ha dividido el proceso de instalación en dos secciones. Una primera para Ubuntu (operativo recomendado) y otra para Mac OS. Los pasos se deben seguir exactamente como figuran en esta guía para asegurar el funcionamiento de las librerías.

Parte 1a. Instalación de SFML junto con Netbeans 8.X en Linux (probado en instalación limpia de ubuntu 16.04 y 17.10)

- Previo instalamos openGL:

```
sudo apt-get install mesa-common-dev  
sudo apt-get install freeglut3-dev
```

- Y el compilador de C++

```
sudo apt-get install build-essential
```

- Instalación de SFML en linux:

```
sudo apt-get install libsFML-dev
```

- ¿Dónde instala los paquetes?

```
sudo dpkg -L libsFML-dev
```

- Instalamos netbeans 8.x desde "Software de Ubuntu"
- Abrimos Netbeans y en **Tools->Plugins->Setting** activamos todos los sources
- En la misma ventana, en "Available Plugins" escogemos el de C++ y lo instalamos
- Reiniciamos netbeans
- Ahora podemos crear un "HolaMundo" en C++ de ejemplo y comprobar que compila

- Crear un proyecto nuevo de C++ de tipo aplicación
- Sobreescribir el main por el siguiente:

```
#include <SFML/Graphics.hpp>

int main()
{
    sf::RenderWindow window(sf::VideoMode(200, 200), "SFML works!");
    sf::CircleShape shape(100.f);
    shape.setFillColor(sf::Color::Green);

    while (window.isOpen())
    {
        sf::Event event;
        while (window.pollEvent(event))
        {
            if (event.type == sf::Event::Closed)
                window.close();
        }

        window.clear();
        window.draw(shape);
        window.display();
    }

    return 0;
}
```

- Ahora configuramos el proyecto para que funcione con SFML. **Nota: este paso se debe hacer para configurar cualquier proyecto de Netbeans con soporte de SFML :**
 - Vamos a las propiedades del Proyecto->"C++ Compiler"
 - En **include directories** ponemos **"/usr/include/SFML"**, verificar, como se ha comentado antes, que en esa ruta se os han instalado los paquetes de la librería
 - En esa misma ventana buscamos **Linker** y ponemos como directorio de librerías adicional la ruta **"/usr/lib/x86_64-linux-gnu"** (verificar como en el paso anterior)
 - En librerías añadir, y en la nueva ventana utilizando el botón **"Add Library File"** debéis incluir **libsFML-window.so**, **libsFML-graphics.so** y **libsFML-system.so** (o sus correspondientes versiones de debug, pero **TODAS DE UN SOLO TIPO**). Estas se deben de encontrar en la ruta del punto anterior.
 - Por último, en la pestaña **Run** escoger como consola una **terminal externa**.
- Al acabar el proyecto de debe poder compilar y ejecutar

¡Enhorabuena, has completado la instalación!

Parte 1b. Instalación en Mac OS X con GCC (probado en 10.12)

- Verifica que tienes instalado gcc. Desde terminal ejecuta **gcc**. Si no está instalado el mismo operativo se encargará de descargarlo. Otra opción es ejecutar **xcode-select**

--install

- Instalamos netbeans 8.x
- Abrimos Netbeans y en **Tools->Plugins->Setting** activamos todos los sources
- En la misma ventana, en "Available Plugins" escogemos el de C++ y lo instalamos
- Reiniciamos netbeans
- Ahora podemos crear un "HolaMundo" en C++ de ejemplo y comprobar que compila
- Descargar SFML para mac en [SFML 2.4.1 \(SFML / Download\)](#)
- Descomprimir el fichero .tar.gz
- Copia el contenido de la carpeta **Frameworks** a **/Library/Frameworks**
- Copia la carpeta **extlibs** a **/Library/Frameworks**.
- Ahora configuramos el proyecto para que funcione con SFML. **Nota: este paso se debe hacer para configurar cualquier proyecto de Netbeans con soporte de SFML :**
 - Vamos a las propiedades del **Proyecto->"C++ Compiler"**
 - Buscamos **Linker** y ponemos en **Additional options:** **-framework sfml-window -framework sfml-graphics -framework sfml-system**

¡Enhorabuena, has completado la instalación!

Parte 2. Prueba la plantilla de juego

Siguiendo los pasos anteriores para la creación de un proyecto Netbeans con soporte para SFML crea uno nuevo (o duplica el anterior) y modifica el **main** del proyecto con el proporcionado en la plantilla. Añade en alguna carpeta accesible del proyecto las imágenes de la plantilla y comprueba su funcionamiento. Contesta a las siguientes preguntas:

- ¿En que ruta se almacena el resultado de la compilación?
- ¿Que rutas son accesibles desde el proyecto?
- ¿Donde has situado las imágenes de la plantilla para poder acceder a ellas?

Parte 3. Implementa el juego retro que te corresponda

Durante el curso 2017/2018 se asignará al azar un videojuego retro distinto para cada uno de los componentes de un grupo ABP. La asignación se realizará durante las primeras sesiones de prácticas. Los videojuegos serán los siguientes:

- Pacman
- Q*bert
- Tron Game [M]
- Joust
- Boinggg
- Tetris
- Puzzle booble
- Flappy Bird
- Galaga
- Micro Machines [M]

Requisitos

Para todos los juegos seleccionados:

- Se recomienda ver un video o jugar en su versión flash en el navegador
- Se deben utilizar las mismas herramientas de desarrollo que para la realización del proyecto ABP
- Por defecto la jugabilidad debe ser la misma aunque se permiten modificaciones siempre y cuando sean aprobadas por el profesor.
- Esta práctica se puede mejorar para optar a 1,5 puntos de parte optativa de la asignatura. Para ello hablar previamente con el profesor.
- Se pueden modificar los gráficos o gastar los originales
- Se debe implementar como mínimo el modo individual. En los juegos señalados con [M] sólo se implementará un modo de 1 jugador contra la máquina.
- No se requiere ni menu de presentación ni pantalla de final de partida. En cuanto nos maten se puede volver a reiniciar el juego.
- El interfaz de entrada será el teclado.
- Como teclas de entrada se gastará el cursor. Si se requiere alguna tecla especial se podrá gastar espacio, control o alt. La tecla 'q' saldrá del juego.
- El juego se ejecutará en ventana y no pantalla completa. Se podrá utilizar la resolución que se considere oportuna para el juego.

Programar el juego

Para empezar a trabajar tienes una plantilla de ejemplo donde se muestran algunas funciones de la librería. Para empezar debes tener en cuenta que:

- La documentación (y varios ejemplos) de SFML se encuentran [aquí](#)
- Posiblemente debas utilizar las siguientes clases de SFML: `sf::Texture`, `sf::Sprite`, `sf::Font`, `sf::Clock`, `sf::Event` y `sf::RenderWindow`
- En cada iteración se debe limpiar la pantalla mediante 'window.clear'

Detalles y entrega

Esta práctica es individual y por tanto **no se permite compartir código ni hacerla en grupo. Se verificará mediante sistema anticopia el código fuente entregado.** Esta práctica es requisito previo al desarrollo del proyecto de la asignatura.

La fecha recomendada de entrega es el 18 de Marzo. Aunque se recomienda respetar esta fecha la entrega permanecerá abierta hasta el día 15 de Abril, no siendo prorrogable dicha fecha.

Apéndice: instalación en otros operativos

Tando el compilador, como las librerías utilizadas son multiplataforma. Sin embargo no se recomienda el uso de un sistema operativo no basado en UNIX, para uniformizar el desarrollo entre los miembros del equipo y para evitar problemas específicos de instalación.

Para preparar el desarrollo del proyecto ABP, todos los miembros del grupo deben, obligatoriamente, gastar el mismo sistema operativo, la misma versión, y las mismas librerías (también de la misma versión). No se recomienda en absoluto actualizarlas mientras que dure el desarrollo del proyecto.

Para instalar y configurar SFML en otros operativos ver [2.4 Tutorials \(SFML / Learn\)](#). Ten en cuenta que el compilador que utilizaremos será **GCC** con lo que no está permitido utilizar otro compilador distinto (no esta permitido utilizar XCODE, Visual Studio...)

La instalación en los laboratorios de SFML está realizada tanto en Ubuntu como en Mac OS.