

# PEC 2 - Un juego de plataformas

Este es un trabajo realizado para la asignatura de Programación 2D del máster en Diseño y Programación de Videojuegos de la UOC.

A continuación se explicarán algunos de los elementos claves del trabajo.

## 1 Recreación del juego original

Para conseguir una sensación lo más fiel al juego original, se buscó en internet el tileset original. Una vez se dividió en paletas y se recreó el mapa a escala 1:1, se ajustó la cámara para tener un ratio de aspecto 4:3 y que se viese sólo lo que se veía en el original y se colocaron los enemigos en sus posiciones adecuadas, teniendo en cuenta la colocación de spawners que evitan que aparezcan antes de tiempo.

Tras buscar en internet, se encontró este artículo que, aun a pesar de ser sobre el Super Mario World, da tanto la velocidad como la gravedad de mario, y la potencia de salto. Resultan ser unas medidas muy próximas al original y que se sienten muy bien.

## 2 Estructura de software del jugador

Para mantener divididos los scripts del jugador, se crean tres diferentes: - PlayerInput - PlayerMovement - PlayerAnimation

El primero, **PlayerInput**, se encarga de mantener actualizadas una serie de variables que se pueden utilizar por los otros scripts, como si se ha pulsado o mantenido el botón de salto, en qué dirección se está queriendo mover el jugador, o si se está pulsando el botón de correr (aunque esta última no está implementada en el movimiento).

**PlayerMovement** utiliza la información del input para decidir qué debe hacer Mario. Cuándo saltar, hacia qué dirección, etc. Además, lleva un control sobre si el jugador ha entrado en contacto con el suelo o con un enemigo desde abajo, y si es el caso, llama a la función Die() de dicho enemigo.

**PlayerAnimation**, por último, implementa animaciones de correr, saltar y morir, y además se encarga de mantener a Mario apuntando al sitio al que

se mueve, a base de invertir el sprite. También es en este script donde se encuentra la función `Die()` de Mario que, aun pudiendo estar en una clase como `PlayerController` o similar, se ha puesto aquí por no crear demasiados scripts y porque, de igual manera, hace falta tener esta función aquí para mostrar la animación de muerte.

### 3 Estructura de software de los enemigos

Los enemigos tienen dos scripts: - Un script de movimiento, que llegado el caso se podría crear uno genérico del que derivasen todos, pero que de momento al haber solo uno, se mantiene ajeno al resto. - Otro script de enemigo, llamado **Enemy**, en el que se halla la función `Die()` y que, en caso de necesitarse, podría heredarse en patrones de comportamiento diferentes (por ejemplo, los koopa no se mueren al caer encima, sino que se esconden).

### 4 Elementos de pantalla

Para conseguir que el jugador no pueda volver atrás, hay una barrera que resulta invisible al jugador pero que se puede ver en el editor. Esta barrera puede avanzar pero no retroceder. Lo mismo sucede con la cámara, que aunque en el editor puede asignársele movimiento libre, normalmente sólo seguirá al jugador sin poder retroceder.

PEC 3

Primero decir que el aspecto 4:3 es a propósito para mantener el mismo campo de visión y pixelart que el juego original. Ahora se explicarán los puntos que se pedían completar en el trabajo:

### 5 Mínimo un personaje con dos o más animaciones

Mario tiene 3 animaciones: Correr, saltar y morir.

Los goombas tienen 2: Andar y morir.

### 6 Mínimo un sistema de partículas en el juego

Al lanzar una bola de fuego, ésta dejará una estela de fuego allá por donde haya pasado que durará poco tiempo.

## **7 Enemigos con un mínimo de inteligencia artificial**

Los goombas avanzan en una misma dirección hasta que chocan contra un obstáculo del entorno, en cuyo momento se dan la vuelta y andan en la dirección contraria.

## **8 Poder disparar a los enemigos**

La bola de fuego de Mario se lanza con el botón E en el teclado. Esta tiene una animación de giro y puede dar un máximo de 4 botes, o desaparece si se choca contra un enemigo, matándolo.

## **9 Control de estado del juego**

Hay 3 vidas al empezar una partida, y una serie de checkpoints a lo largo del juego. Si llegas a un checkpoint más avanzado que el último, este será sustituido por el nuevo. Si mueres pero te siguen quedando vidas, reaparecerás en uno de estos checkpoints.

## **10 Uso de tags y layers**

Las tags y layers se usan para evitar que, por ejemplo, los goombas choquen entre ellos, o que las bolas de fuego choquen con el propio Mario o entre ellas.