

Super Mario World Clone

Curso 2k21/2k22

Apellidos, nombre	Hernández Rocha, Airam (aiherroc@inf.upv.es) Alejandro Moreno González (almogon3@inf.upv.es)
Titulación	Grado de Ingeniería informática
Fecha	Abril 2022

Índice

Resumen de las ideas clave	4
Introducción	4
Objetivos	5
Planificación temporal	5
Desarrollo	6
5.1. Estructura de la aplicación	6
Mario	6
Toad	7
ToadText	7
Goomba	7
Coin	7
Button	7
Block	7
Title, Scoreboard, TimeState, Background y Credits	8
5.2. Lógica implementada	8
Conclusión	10
Trabajo futuro	10
Bibliografía	11
Sprites:	11
Audio:	11
Anexo de commits	12

Índice de figuras

Figura 1: Diagrama de Gantt	5
Figura 2: Pantalla de juego en ejecución	8
Figura 3: Pantalla de Menú	9

1 Resumen de las ideas clave

En esta memoria se presentará el proceso y resultados del desarrollo mediante las herramientas de DevkitPro para la plataforma Nintendo 3DS, de un programa de imitación del videojuego "Super Mario World" desarrollado por Nintendo y lanzado al mercado el 21 de noviembre de 1990.

Los esfuerzos se han centrado en la recreación de los movimientos principales del personaje, así como la interacción con los elementos más comunes presentes en el título.

2 Introducción

Como proyecto hemos desarrollado un clon del juego de Mario 2D "Super Mario World", que sirve como primeros pasos para poder el desarrollo de una versión completa del mismo. Por lo que hemos implementado las bases mecánicas, y dejado a futuro la creación de nuevos mapas y contenido (niveles, enemigos y *power ups*).

Una de las ventajas de afrontar este tipo de proyecto es que hay una gran cantidad de recursos disponibles de forma abierta, como son los *sprites*¹ de los personajes y escenarios, (aunque para algunos elementos se han tenido que crear desde 0) o la banda sonora² original.

En resumen, se ha querido desarrollar una base jugable de los elementos principales que definen a la franquicia de Nintendo, por lo que el movimiento del personaje principal y sus animaciones son las protagonistas del proyecto.

3 Objetivos

Una vez finalizado el desarrollo, los siguientes requisitos deben estar implementados:

- Mario debe tener su movimiento 2D básico, incluyendo saltos y agacharse.
- Mario al ser golpeado debe transformarse en su forma en miniatura.
- Durante toda la duración del programa, debe estar sonando la banda sonora del primer mundo original.
- Deben haber efectos de sonido de Mario al saltar³, del bloque al ser golpeado con la moneda⁴, del Goomba al morir⁵ y de Toad al hablar con él⁶, todos lo más parecido posible al de la obra original.
- En la pantalla inferior debe mostrarse el contador de vidas restantes, el contador de monedas y el tiempo de la partida actual.
- El escenario de fondo debe ser similar al del primer mundo del juego original.

4 Planificación temporal

Se ha diseñado el siguiente diagrama de Gantt para mostrar como fue el diseño de la aplicación, al principio realizamos la obtención y diseño (de aquellos no disponibles) de los sprites, el mapa y las animaciones.

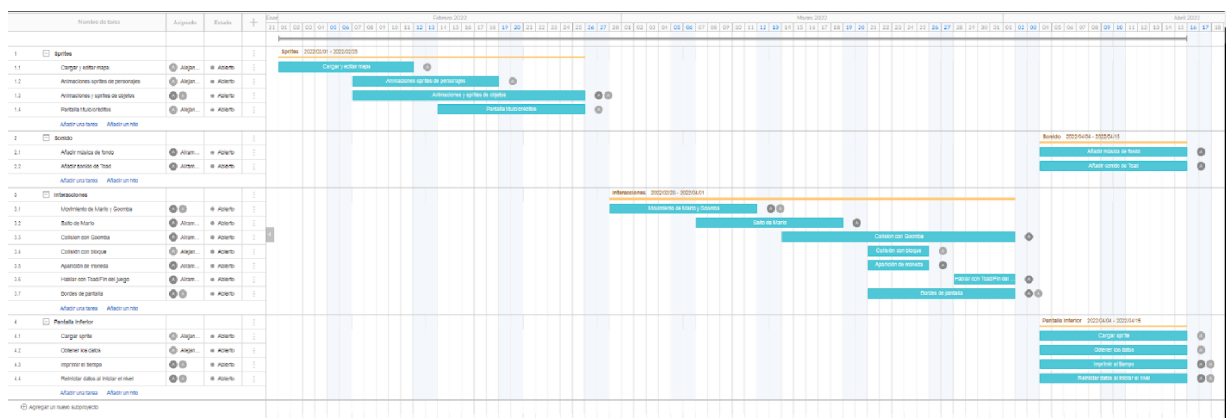


Figura 1: Diagrama de Gantt

Los apartados más costosos fueron el desarrollo de las colisiones y las interacciones con los objetos. La parte final fue enfocada al diseño de la pantalla inferior.

5 Desarrollo

Para el desarrollo se han utilizado las herramientas vistas en clase incluidas en la instalación de DevkitPro. Entre ellas, la principal es "Citro2D", ya que se encarga de todas las cuestiones gráficas. Para el apartado del audio, los archivos de sonido se ha elegido utilizarlos en formato ".cwav", por lo que la librería encargada es "libcwav".

5.1. Estructura de la aplicación

El proyecto se compone de dos archivos, uno principal escrito en C++ para la lógica de la aplicación, y un archivo de headers que define todas las constantes y estructuras a utilizar, entre las que se encuentran el propio personaje principal, el enemigo o el bloque.

Como anexo, se presenta además el historial de *commits* llevados a cabo, que reflejan el proceso de desarrollo. Se dividen en dos repositorios diferentes debido a problemas que finalmente no han tenido mayor impacto en el proyecto.

Al tratarse de un videojuego, se ha decidido crear una estructura de datos para cada elemento en pantalla. En el resultado final, se ha acabado con las siguientes:

5.1.1. Mario

Almacena toda la información necesaria para el control del jugador, la interacción con el resto de elementos y el estado actual y datos necesarios para las animaciones del personaje.

Podemos destacar entre ellos cosas como la posición en los ejes x e y, el sprite actual, la velocidad, monedas, vidas, el estado actual (caminando,

saltando, cayendo o muerto), las animaciones, o límites como el máximo tiempo de salto.

5.1.2. Toad

Como su único propósito es proporcionar un diálogo, solamente se almacena su animación y sprite actual.

5.1.3. ToadText

Se trata del diálogo que se activará cuando se interactúe con Toad. Maneja su visibilidad y almacena su sprite. Se podría extender para contener diferentes sprites con otros textos.

5.1.4. Goomba

Almacena la posición actual, el sprite utilizado y animaciones, hacia dónde se dirige, la moneda que posee, su rotación actual (para la animación de muerte) y velocidad.

5.1.5. Coin

Almacena su visibilidad y los datos necesarios para su animación (posición, tiempo y sprite).

5.1.6. Button

Estructura que representa la imagen del botón a pulsar para las interacciones de diálogo y el menú principal.

5.1.7. Block

Representa el bloque a romper, para lo que solamente necesita su posición, su estado (roto o no), el sprite actual y la moneda que almacena.

5.1.8. Title, Scoreboard, TimeState, Background y Credits

Se trata de estructuras secundarias necesarias para el funcionamiento del HUD de usuario, menú o elementos estáticos del escenario.

5.2. Lógica implementada

Lo primero que se llevó a cabo en el desarrollo fue el movimiento del personaje en el eje X, para lo cual se utilizan listas que contienen los *sprites* que conforman cada animación. No se sustituye el *sprite* en cada frame, ya que entonces no daría la sensación de que está caminando de forma natural, sino que se deja un margen de milisegundos entre cada cambio.

La acción de saltar se divide en dos fases, el propio salto o elevación, y la caída. Se establece un máximo de milisegundos que el jugador puede mantener pulsado (durante los cuales Mario se eleva), y una vez se suelta el botón o se alcanza este tope, se inicia la caída y cambia el *sprite* del personaje.



Figura 2: Pantalla de juego en ejecución

La banda sonora se reproduce en formato “.wav” en bucle durante toda la ejecución, mientras que los efectos de sonido se activan en los momentos en los que son necesarios. Sin embargo, todos ellos se cargan de forma estandarizada al inicio del programa.

Las colisiones son manejadas mediante los datos de posición presentes en todos los objetos que toman parte, como son Mario, el bloque y el Goomba. Se calcula si sus posiciones coinciden, ajustando con los márgenes de tamaño provocados por el alto y ancho de los propios sprites, y se determina qué interacción se debe llevar a cabo, tomando en cuenta además, el estado de los objetos implicados. Es decir, una vez colisionan Mario y el Goomba por sus posiciones, si el estado de Mario indica que está saltando o cayendo, acabará con el enemigo, que hará su animación y muerte y proveerá una moneda. En caso contrario (si Mario está caminando), o bien se altera el estado de Mario y sus animaciones para que se correspondan con las de su versión en miniatura, o en caso de ya estar en dicha versión, se le quitará una vida y dará un nuevo intento.

Una vez se agotan las vidas, el menú principal vuelve a mostrarse y se deberá empezar de nuevo.

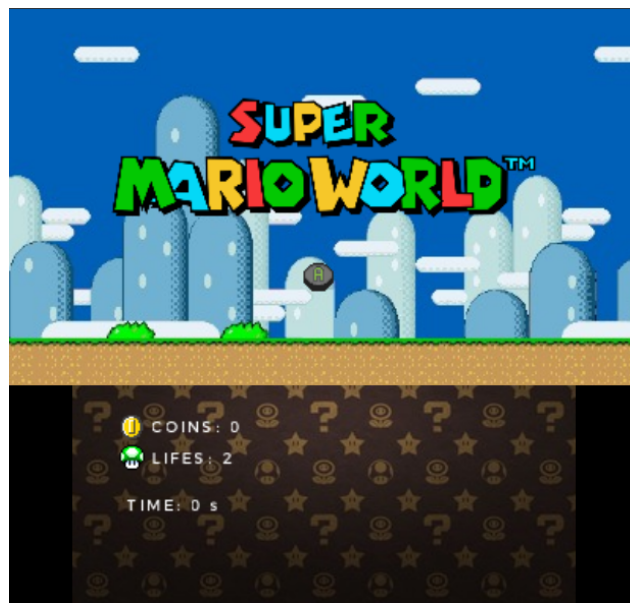


Figura 3: Pantalla de menú

Para terminar la partida se debe hablar con el Toad que se encuentra en la parte derecha de la pantalla, que se encuentra en una animación en bucle de llanto. Tanto el indicador del botón a pulsar (que también aparece en el menú) como el propio diálogo son *sprites* creados desde cero.

6 Conclusión

Durante el desarrollo hemos podido comprobar la viabilidad de las herramientas presentes en DevkitPro a la hora de desarrollar un videojuego en dos dimensiones y ello nos ha dejado varias lecciones. En primer lugar, pese a no tratarse del SDK oficial de desarrollo para la plataforma, sí que posee la suficiente potencia como para poder utilizarse para proyectos de más envergadura.

Sin embargo, es posible que si se desarrollara un *framework* que funcionara de abstracción para Citro2D, Citro3D, libctru o libcwav, se atraería mucha más atención y comunidad, lo que ayudaría también con la documentación presente actualmente en internet, y principalmente, abriría las puertas a proyectos más ambiciosos en menor tiempo de desarrollo. Está claro que este set de herramientas es un proyecto comunitario que no cuenta con la visibilidad ni el público objetivo suficiente para este tipo de desarrollos, pero por proponer que no quede.

7 Trabajo futuro

Los pasos siguientes si se quisiera completar este proyecto creemos que deberían ser los siguientes:

1. Desarrollar una abstracción sobre Citro2D y las librerías principales de DevkitPro para hacer que el desarrollo fuera mucho más escalable, estandarizado y menos tedioso.
2. Implementar el movimiento de la cámara junto con el jugador, pudiendo así crear niveles más grandes que el tamaño de la propia pantalla.

3. Crear un sistema de niveles y mundos, además de una pantalla de selección de los mismos
4. Implementar los potenciadores y enemigos restantes para igualar la experiencia original.

8 Bibliografía

Bibliografía y referencias, desglosadas o no en apartados.

8.1. Sprites:

- [1] Mario Universe.com, Super Mario World SpriteSheets - NES.
<http://www.mariouniverse.com/sprites-snes-smw/>

8.2. Audio:

- [2] Overworld Theme - Super Mario World.
<https://www.youtube.com/watch?v=tAaGKo4XVvM&t=5s>
- [3] Mario Jump - Gaming Sound Effect.
<https://www.youtube.com/watch?v=37-paiEz0mQ>
- [4] Super Mario Bros. Coin Sound Effect.
<https://www.youtube.com/watch?v=Rfkcl8dhfsQ>
- [5] Goomba Death Sound Effect. <https://www.youtube.com/watch?v=EVQTJ5Xrw0g>
- [6] Toad Death Sound Effects. <https://www.youtube.com/watch?v=mA-JO0Q†CA0>

airamhr9 / aev_mario Private

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#)

main

Commits on Apr 25, 2022

Mario se agacha

 airamhr9 committed 18 minutes ago

Commits on Apr 24, 2022

Iconos hud


 airamhr9 committed yesterday

Todos los efectos de sonido puestos


 airamhr9 committed yesterday

Commits on Apr 22, 2022

Sistema de vidas

 airamhr9 committed 3 days ago

mejor animación de daño


 airamhr9 committed 3 days ago

animacion daño


 airamhr9 committed 4 days ago

Commits on Apr 21, 2022


solucionado bug el tiempo no avanza

 airamhr9 committed 4 days ago

Hecho todo lo relacionado con la pantalla de fin

 alecahy committed 5 days ago

Cambios en la pantalla de fin

 alecahy committed 5 days ago

Commits on Apr 20, 2022

Añadido sprite de pantalla final



airamhr9 committed 5 days ago

Ahora cuenta bien las vidas



alecahy committed 6 days ago

🔗 Commits on Apr 19, 2022

Código para arrancar la ventana de fin



alecahy committed 6 days ago

refactor y bug fixes de la pantalla inferior



airamhr9 committed 6 days ago

Bottom Score Update



alecahy committed 6 days ago

Avances con la obtención de la ventana inferior



alecahy committed 6 days ago

Cambiado el fondo de la pantalla de abajo



alecahy committed 6 days ago

🔗 Commits on Apr 17, 2022

Añadida imagen inferior



airamhr9 committed 8 days ago

bug fix



airamhr9 committed 8 days ago

Mario chiquito



airamhr9 committed 8 days ago

Empiezo la pantalla de puntuación de abajo



alecahy committed 8 days ago

🔗 Commits on Apr 16, 2022

Pantalla de abajo - Inicio



alecahy committed 9 days ago

🔗 Commits on Apr 15, 2022

Aparece menu tras muerte



airamhr9 committed 10 days ago

Bug fixes



airamhr9 committed 10 days ago

Muerte de mario con animación

airamhr9 committed 10 days ago

- Commits on Apr 14, 2022

Menú inicial

airamhr9 committed 11 days ago

- Commits on Apr 13, 2022

Añadida moneda a bloque y goomba

airamhr9 committed 12 days ago

Update .gitignore

airamhr9 committed 12 days ago

Delete aev_mario.smdh

airamhr9 committed 12 days ago

Delete aev_mario.elf

airamhr9 committed 12 days ago

Delete aev_mario.3dsx

airamhr9 committed 12 days ago

Delete build directory

airamhr9 committed 12 days ago

colisiones y muerte goomba

airamhr9 committed 12 days ago

- Commits on Apr 12, 2022

Colisiones con Goomba y bordes de la pantalla

alecahy committed 13 days ago

- Commits on Apr 10, 2022

Colisiones con bloque

airamhr9 committed 15 days ago


Colision inferior con bloque

airamhr9 committed 15 days ago

[Newer](#)[Older](#)


 alecahy / **Super_Mario_64_DS_Clone** Private


 **Code**  Issues  Pull requests  Actions  Projects  Security 

 mario_model ▾

 Commits on Apr 8, 2022


Banda sonora primera implementacion

 airamhr9 committed 18 days ago

 Commits on Mar 26, 2022

Salto y caminar hecho

 airamhr9 committed on 26 Mar

 Commits on Mar 21, 2022

Mario se mueve bien

 airamhr9 committed on 21 Mar

Cambiado mapa y añadido bloques

 airamhr9 committed on 21 Mar


Merge branch 'mario_model' of https://github.com/alecahy/Super_Mario_64_DS_Clone ...

 airamhr9 committed on 21 Mar

Deleted Super Mario 64 Ds Clone subdir

 airamhr9 committed on 21 Mar

Delete build directory

 airamhr9 committed on 21 Mar

Delete Super_Mario_World_Clone.smdh

 airamhr9 committed on 21 Mar

Delete Super_Mario_World_Clone.elf

 airamhr9 committed on 21 Mar

Delete Super_Mario_World_Clone.3dsx

 airamhr9 committed on 21 Mar

Delete .vscode directory

 airamhr9 committed on 21 Mar

Delete .vs directory



airamhr9 committed on 21 Mar

Super Mario World con la programación del bloque



alecahy committed on 21 Mar

Commits on Mar 7, 2022

mejor movimiento de mario y consola para debug



airamhr9 committed on 7 Mar

Commits on Mar 1, 2022

Mario moves



airamhr9 committed on 1 Mar

Commits on Feb 26, 2022

Mario animations and movements



alecahy committed on 26 Feb

Commits on Feb 23, 2022

Remake 2D



airamhr9 committed on 23 Feb

Commits on Feb 21, 2022

Añadido modelo de mario roto



airamhr9 committed on 21 Feb

Añadido modelo de mario roto



airamhr9 committed on 21 Feb

Commits on Feb 20, 2022

Archivos del proyecto



alecahy committed on 20 Feb

[Newer](#)[Older](#)