

ADRIÁN TIENE HAMBRE

María Rosario López Soler

Grado en Multimedia

TFG - Videiuegos

Consultor: Joel Servitja Feu

Profesor responsable de la asignatura: Javier Luis Cánovas Izquierdo

Enero 2020



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Adrián tiene hambre</i>
Nombre del autor:	<i>M^a Rosario López Soler</i>
Nombre del consultor/a:	<i>Joel Servitja Feu</i>
Nombre del PRA:	<i>Javier Luis Cánovas Izquierdo</i>
Fecha de entrega (mm/aaaa):	12/2019
Titulación:	<i>Grado en Multimedia</i>
Área del Trabajo Final:	<i>TFG Videojuegos</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Unity, plataformas, coleccionables</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i>	
<p>“Adrián tiene hambre” es un juego de plataformas 2D para PC inspirado en los clásicos arcade. El jugador debe recoger biberones en un escenario simple donde hay plataformas y enemigos. El objetivo del proyecto ha sido crear este videojuego pasando por todas las fases, esto es, desde la planificación y diseño de los diferentes elementos gráficos hasta la programación para implementar el juego con el fin de obtener una versión final jugable.</p> <p>La metodología ha consistido principalmente planificar el trabajo y en la ejecución de las diferentes tareas, ya que me ello me ha ayudado a organizar un calendario de seguimiento. Al tratarse de mi primer trabajo con Unity he tenido que hacer pequeños reajustes temporales debido al tiempo que he necesitado en aprender a manejar este motor de videojuegos.</p> <p>A pesar de la limitación temporal para llevar a cabo este trabajo estoy contenta con lo que he conseguido hacer, sobre todo, por los conocimientos que he adquirido sobre Unity y por todo lo que se podría hacer en un futuro.</p>	
Abstract (in English, 250 words or less):	
<p>"Adrián is hungry" is a 2D platform game for PC inspired in arcade classic games. The player must collect baby bottles in aN easy scenario where there are platforms and enemies. The objective of the project is to create this videogame through all the phases, from the planning and design of the different graphic elements to the programming to implement the game in order to obtain a final playable version.</p> <p>The methodology has mainly consisted of planning the work and the execution</p>	

of the different tasks; it helped me to organize a follow-up calendar. As it was my first job with Unity, I had to make small temporary adjustments due to the time I needed to learn how to drive this video game engine.

In spite of the temporary limitation to carry out this work, I am happy with what I have managed to do, above all, for the knowledge I have acquired about Unity and for everything that could be done in the future.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	1
1.3 Enfoque y método seguido.....	1
1.4 Planificación del Trabajo.....	2
1.5 Breve resumen de productos obtenidos.....	2
1.6 Breve descripción de los otros capítulos de la memoria.....	3
2. Estado del arte.....	4
2.1 Género del juego.....	4
2.2 Plataformas de desarrollo.....	4
3. Definición del juego.....	6
3.1 Historia y ambientación.....	6
3.2 Objetivos planteados al jugador.....	6
3.3 Definición de los personajes y elementos.....	6
3.4 Tipo de interacción juego-jugador.....	7
3.5 Plataforma de destino.....	7
3.6 Concept Art.....	7
4. Diseño técnico.....	11
4.1 Entorno de desarrollo escogido y requerimientos técnicos.....	11
4.2 Inventario de herramientas empleadas.....	11
4.3 Assets y recursos del juego.....	12
4.4 Esquema de arquitectura del juego.....	18
4.5 IA de los enemigos.....	18
5. Diseño de niveles.....	21
6. Manual de usuario.....	22
6.1 Requerimientos técnicos del hardware.....	22
6.2 Instrucciones del juego.....	22
7. Conclusiones.....	23
8. Glosario.....	25
9. Bibliografía.....	26

Lista de figuras

Ilustración 1: Diagrama de Gantt	2
Ilustración 2: Videojuego Mario Bros Classic	4
Ilustración 3: Videojuego Donkey Kong	4
Ilustración 4: Logo Unity	4
Ilustración 5: Logo GameMaker Studio	5
Ilustración 6: Logo GDevelop	5
Ilustración 7: Ejemplo de sprite sheet de bebé gateando	7
Ilustración 8: Pantalla Menú	8
Ilustración 9: Pantalla Intro	8
Ilustración 10: Pantalla de juego	8
Ilustración 11: Pantalla Game Over	9
Ilustración 12: Pantalla Pause	9
Ilustración 13: Pantalla final si se ha conseguido la medalla de oro	9
Ilustración 14: Pantalla final si se ha conseguido la medalla de plata	10
Ilustración 15: Pantalla final si se ha conseguido la medalla de bronce	10
Ilustración 16: Requerimientos técnicos para desarrollo con Unity 2019.2	11
Ilustración 17: Fondo del videojuego	14
Ilustración 18: Capa 7 - cielo	14
Ilustración 19: Capa 6 - montañas	14
Ilustración 20: Capa 5 - nubes	14
Ilustración 21: Capa 4 – colinas3	14
Ilustración 22: Capa 3 – colinas2	14
Ilustración 23: Capa 2 – colinas1	14
Ilustración 24: Capa 1 - suelo	15
Ilustración 25: Capa nueva del cielo	15
Ilustración 26: Personaje principal del juego	15
Ilustración 27: Sprite gateo izquierda	16
Ilustración 28: Sprite gateo derecha	16
Ilustración 29: Idle derecha	16
Ilustración 30: Idle izquierda	16
Ilustración 31: Saltos derecha e izquierda	16
Ilustración 32: Enemigos del videojuego	17
Ilustración 33: Creación y sprite del coleccionable	17
Ilustración 34: Música de fondo del videojuego	17
Ilustración 35: Pistas de audio para los efectos sonoros	17
Ilustración 36: Idle del enemigo 1	18
Ilustración 37: Walking enemigo 2	19
Ilustración 38: Running enemigo 2	19
Ilustración 39: Idle enemigo 3	19
Ilustración 40: Walking enemigo 3	19
Ilustración 41: Running enemigo 3	20
Ilustración 42: Mapa del nivel del juego	21
Ilustración 43: Instrucciones del juego	22

1. Introducción

1.1 Contexto y justificación del Trabajo

El mundo de los videojuegos ha experimentado un notable cambio a lo largo del tiempo. Hace algunas décadas solo se podía jugar en establecimientos recreativos ante enormes máquinas arcade que trataban de saciar la necesidad de entretenimiento de las generaciones más jóvenes. Con la llegada de las primeras consolas portátiles se hizo posible trasladar dicho entretenimiento al salón de casa, un importante fenómeno social que inició una nueva forma de diversión que persiste hoy en día.

La industria de los videojuegos está creciendo a grandes pasos porque existe una gran demanda del mercado. Ello hace que se creen puestos de trabajo en este ámbito y, cada vez más, se necesitan profesionales de diferentes perfiles (diseñadores gráficos, animadores, *game designers*, programadores, etc.) para poder llevar a cabo de forma conjunta la creación de un producto audiovisual de estas características.

Los avances tecnológicos e informáticos, junto al auge de Internet, han hecho posible que actualmente cualquier persona pueda crear su propio videojuego, lo cual era impensable en otros tiempos. Es por ello que en este Trabajo he querido crear uno propio por primera vez y desde cero para conocer más de cerca cómo es este proceso.

1.2 Objetivos del Trabajo

Este trabajo ha consistido en crear un videojuego 2D de plataformas para PC. Para ello, los objetivos han sido los siguientes:

- Aprender a utilizar el motor de videojuegos Unity.
- Planificar el proyecto desde el inicio hasta el final.
- Buscar y crear los elementos gráficos y sonoros necesarios para montar el escenario (diseño del nivel).
- Tener una gran motivación por crear un videojuego propio sin tener ninguna experiencia previa.

1.3 Enfoque y método seguido

La creación de un videojuego requiere una inversión tanto de tiempo como de recursos humanos (diseñadores gráficos, programadores, *game designers*, etc.). Debido al poco tiempo disponible para llevar a cabo el proyecto y mi nula experiencia en la utilización de Unity, he optado por realizar un producto nuevo, pero sencillo con unos controles fáciles.

En la parte gráfica he optado por buscar en fuentes gratuitas de Internet la mayoría de los assets. Solamente he comprado uno, el personaje principal, ya que hacer un diseño propio habría consumido mucho tiempo. Ello me ha permitido disponer de ese tiempo para aprender a manejar el programa Unity.

Respecto a la parte de elementos sonoros también he accedido a fuentes de la red, ya que actualmente existen numerosas páginas con miles de efectos sonoros. Una vez obtenidos los que me interesaban, con Audacity he podido realizar pequeñas modificaciones.

En cuanto a la metodología he seguido las siguientes fases:

- Pensar en el tipo de juego que se quiere hacer.
- Elección del programa para desarrollar el juego.
- Búsqueda y/o creación de assets y recursos.
- Diseño del nivel.
- Diseño de las animaciones.
- Programación.
- Pruebas.
- Correcciones.
- Obtener el producto final.

1.4 Planificación del Trabajo

Para realizar el trabajo se han necesitado los siguientes recursos:

- Creación del videojuego: programa Unity.
- Obtención de assets y sprites: internet y photoshop.
- Obtención de archivos sonoros: internet.

El siguiente diagrama muestra una planificación detallada de las tareas a llevar a cabo.

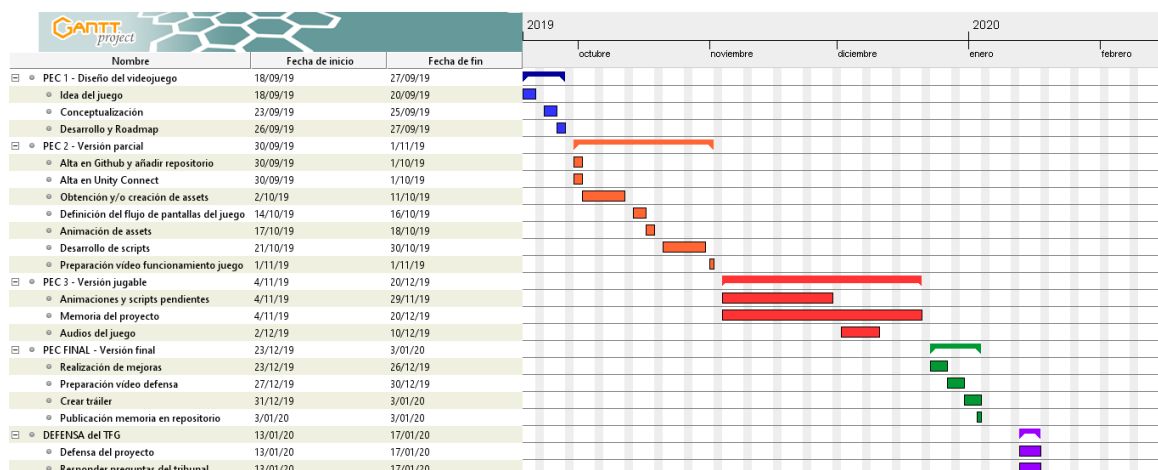


Ilustración 1: Diagrama de Gantt

1.5 Breve resumen de productos obtenidos

Los productos obtenidos a lo largo del curso han sido:

- Vídeos de las PEC 2 y 3.
- Fichero ejecutable con la versión Alpha del videojuego.

- Fichero ejecutable con la versión final del videojuego.
- Diagrama de Gantt con la planificación del proyecto.
- Showcase de Unity Connect.
- Archivos del desarrollo del trabajo (código fuente).
- Vídeo de defensa del proyecto.
- Informe de autoevaluación.
- Memoria del proyecto.

1.6 Breve descripción de los otros capítulos de la memoria

- **Capítulo 2:** Revisión sobre el género del juego y revisión de plataformas de desarrollos que hay en el mercado para hacer juegos.
- **Capítulo 3:** Descripción de la historia y ambientación, objetivos planteados al jugador, definición de los distintos componentes (personajes y elementos) y Concept Art.
- **Capítulo 4:** justificación del entorno de desarrollo escogido, inventario de las herramientas que se han utilizado para hacer el proyecto y recursos gráficos y sonoros.
- **Capítulo 5:** Descripción de cómo se ha creado el nivel del juego.
- **Capítulo 6:** Manual de usuario con los requerimientos técnicos e instrucciones del juego.
- **Capítulo 7:** Conclusiones del proyecto.
- **Capítulo 8:** Glosario con los términos más relevantes
- **Capítulo 9:** Bibliografía.

2. Estado del arte

2.1 Género del juego

Se basa en el género de plataformas en 2D tipo arcade. Un videojuego arcade se caracteriza por tener un diseño sencillo y controles fáciles. También, los niveles son cortos para que el jugador tenga ganas de seguir jugando y, además, la dificultad del juego va ascendiendo a medida que avanza.

Mi idea es seguir los principios básicos de los juegos clásicos de las décadas 80 y 90. La lista de videojuegos es muy extensa, pero algunos de los más conocidos de esa época fueron: Pac-Man, Tetris, Street Fighter, Donkey Kong, Arkanoid y Super Mario Bros, que eran los que más me gustaban. He querido coger como referencia “Super Mario Bros Classic” para poder crear y desarrollar “Adrián tiene hambre”.



Ilustración 2: Mario Bros Classic

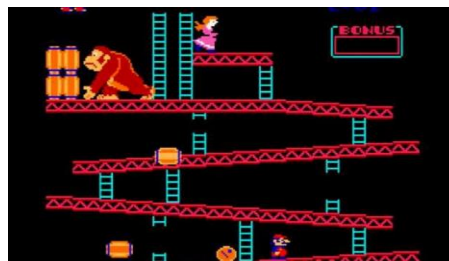


Ilustración 3: Donkey Kong

2.1 Plataformas de desarrollo

Actualmente existen muchos engines para desarrollar juegos 2D. Los tres más conocidos para desarrollar juegos 2D son:



Ilustración 4: logotipo de Unity

Unity es uno de los motores de videojuegos 2D y 3D más conocidos hoy en día. La amplia documentación con la que cuenta, además de foros y otras opciones disponibles para estar en contacto con comunidades afines, hacen que sea una herramienta muy utilizada en todo el mundo tanto por profesionales como por particulares. Además, permite crear aplicaciones para varios tipos de plataformas (Android, PC, etc.).

La IU del programa es intuitiva una vez que te familiarizas con las distintas ventanas. Utiliza C# como lenguaje de programación, que no es difícil, por lo que cualquier persona puede crear su propio videojuego en poco tiempo.

Unity dispone de varios planes, pero hay uno gratuito para los programadores independientes siempre y cuando sus ingresos sean inferiores a los 100.000 \$ en los últimos 12 meses, por lo que se trata de una plataforma básica de desarrollo en tiempo real excepcional para iniciarse en el mundo de los videojuegos.



Ilustración 5: Logo GameMaker Studio

GameMaker Studio es una plataforma basada en un lenguaje de programación interpretado y un kit de desarrollo de software para desarrollar videojuegos [1]. Dicho lenguaje es propio, llamado GML (GameMaker Language), lo que hay que aprenderlo desde cero. Su editor permite *drag and drop* (arrastrar y soltar) con los objetos y otros componentes. Al igual que Unity también permite exportar a diferentes tipos de plataformas existentes en el mercado.



Ilustración 6: Logo GDevelop

GDevelop es una plataforma de código abierto para crear juegos en 2D. Se puede descargar de forma gratuita y es fácil de utilizar. Es posible hacer cualquier tipo de juego: plataformas, rompecabezas, estrategia, etc. Una característica muy destacable es que no es necesario aprender ningún lenguaje de programación, ya que su funcionamiento se basa en un sistema de eventos intuitivos (programación visual). Los juegos creados con este engine pueden publicarse en iOS, Android, Windows, Mac y Linux, entre otros

3. Definición del juego

3.1 Historia y ambientación

Los padres de Adrián están fuera de casa y no se sabe a qué hora volverán, por lo que la madrina del bebé, su tía María, está a cargo de su cuidado. Mientras que el pequeño está durmiendo en su cuna, María habla por teléfono con una amiga. Adrián se ha despertado y se queja porque tiene hambre, pero como María no se ha dado cuenta de ello, el bebé consigue salir de la cuna y empieza a gatear por la casa. Pasa por delante de la puerta del salón y ve a su tía concentrada en su conversación telefónica, por lo que decide ir a la cocina. Una vez allí toca la puerta de la nevera y, de repente, aparece en un lugar fascinante (el juego) donde puede recoger biberones.

3.2 Objetivos planteados al jugador

El objetivo del jugador es recorrer el escenario de izquierda a derecha a través de plataformas para recoger los biberones. Para ello debe utilizar las habilidades del personaje (gatear y saltar) y tener mucho cuidado porque si cae a la parte inferior del escenario (zona de muerte) o si alguno de los malos consigue raptarlo tendrá que empezar de nuevo la partida. El juego será superado si el jugador consigue llegar al final del nivel habiendo recogido un mínimo de 40 biberones.

3.3 Definición de los personajes y elementos

Personaje principal

Es un bebé de 9 meses que acaba de aprender a gatear. Es muy avisado y listo, se fija en todo con detenimiento como si fuera una persona adulta y, además, tiene un don especial para detectar peligros a su alrededor.

Enemigos

Son raptores de bebés. Habrá tres personajes de este tipo a lo largo del nivel con diferentes tipos de movimientos: idle (quieto), caminando y corriendo. El bebé solo tiene que esquivarlos sin tocarlos, ya que si los toca perderá la partida.

Elementos que aparecen en el escenario

- **Biberones:** son los únicos elementos coleccionables del juego que sirven también para guiar al jugador para que sepa por dónde ir.
- **Fondo del videojuego:** formado por montañas, nubes y árboles.
- **Vegetación:** hierba, setas, cactus, árboles.
- **Cajas y barriles** de madera.
- **Vallas** de madera.
- **Medallas:** bronce, plata y oro.

- **Piedra.**
- **Troncos** fijos y móviles.
- **Plataformas** fijas y móviles.
- **Puerta** de fin de nivel.

3.4 Tipo de interacción juego-jugador

El jugador interactúa en el juego a través de los controles establecidos. En este caso se hará uso, por un lado, del ratón del ordenador para poder pulsar sobre los botones de las distintas pantallas de navegación y, por otro lado, del teclado para poder usar las habilidades del personaje. Las teclas son las siguientes:

- **Cursor flecha derecha:** gatear hacia la derecha
- **Cursor flecha izquierda:** gatear hacia la izquierda
- **Tecla z:** saltar
- **Tecla esc:** pausa

3.5 Plataforma de destino

La plataforma de destino es solamente el PC debido a que se trata del primer videojuego que desarrollo.

3.6 Concept Art

En la fase inicial del proyecto, concretamente en la definición del juego, busqué en internet imágenes de bebés para poder hacerme una idea de cómo podría ser el protagonista del juego.



Ilustración 7: ejemplo de sprite sheet de bebé gateando [2]

No he llegado a realizar bocetos, ya que me fui inspirando en muchas imágenes de Internet para poder después crear el escenario directamente, pero poco a poco.

Llegados a este punto sí tenía claro cuáles serían las pantallas de navegación del juego, aunque posteriormente han aumentado (he añadido: Pause e Intro). A continuación se muestra el flujo de navegación:



Ilustración 8: Pantalla Menú



Ilustración 9: Pantalla Intro

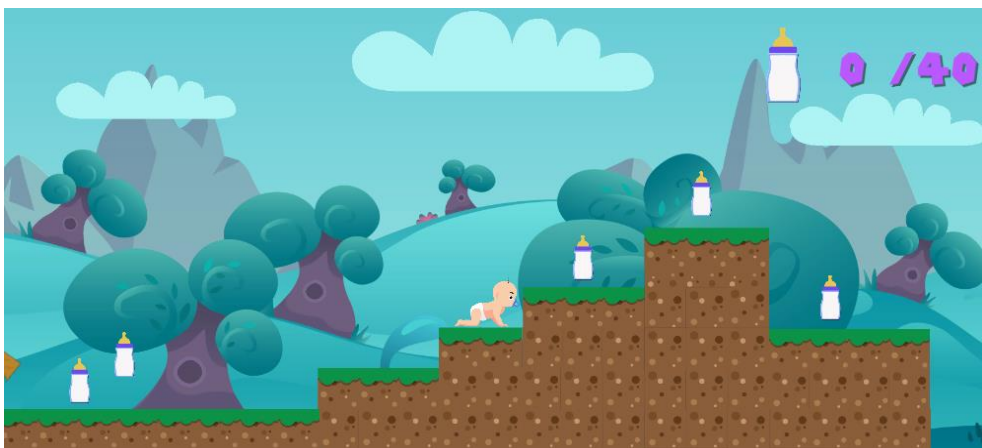


Ilustración 10: Pantalla de juego

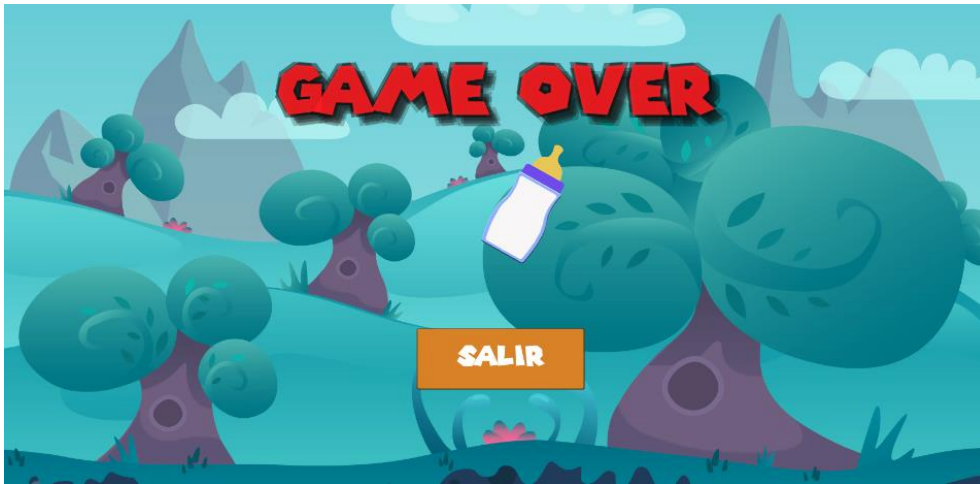


Ilustración 11: Pantalla Game Over



Ilustración 12: Pantalla Pause



Ilustración 13: Pantalla final si se ha conseguido la medalla de oro



Ilustración 14: Pantalla final si se ha conseguido la medalla de plata



Ilustración 15: Pantalla final si se ha conseguido la medalla de bronce

4. Diseño técnico

4.1 Entorno de desarrollo escogido y requerimientos técnicos

He utilizado Unity porque se trata de un motor gráfico muy utilizado. Aunque partía de cero en cuanto a desarrollo de videojuegos, hacía mucho tiempo que quería hacer uso de este programa, por lo que el TFG ha sido una buena oportunidad para ello. Además, Unity es gratuito, dispone de mucha documentación y el lenguaje de programación C# me es familiar.

En cuanto a los requerimientos técnicos mínimos para desarrollar un producto con Unity, en su página web oficial informa que son los siguientes para la versión 2019.2:

System	Minimum requirements
Operating system	Windows: 7 SP1+, 8, 10, 64-bit versions only macOS: 10.12+ Linux: Fixed at: Ubuntu 16.04, 18.04 and CentOS 7 Server versions of Windows and OS X are untested.
CPU	SSE2 instruction set support.
GPU	Graphics card with DX10 (shader model 4.0) capabilities.
Devices	
iOS	Mac computer running minimum macOS 10.12.6 and Xcode 9.4 or higher.
Android	Android SDK and Java Development Kit (JDK). IL2CPP scripting backend requires Android NDK.
Universal Windows Platform	Windows 10 (64-bit), Visual Studio 2015 with C++ Tools component or later and Windows 10 SDK.

Ilustración 16: Requerimientos técnicos para desarrollo con Unity 2019.2

4.2 Inventario de herramientas empleadas



UNITY 2018.1.6f1

Herramienta principal para la creación y desarrollo del videojuego.



ADOBE PHOTOSHOP CS5

Herramienta empleada para crear y modificar los diseños gráficos del videojuego (sprites, etc.).



AUDACITY

Programa de edición de audio gratuito empleado para realizar pequeñas modificaciones en los archivos sonoros.



OBS STUDIO

Programa de grabación de audio e vídeo utilizado para crear los distintos vídeos entregados a lo largo del semestre.



VISUAL STUDIO 2017

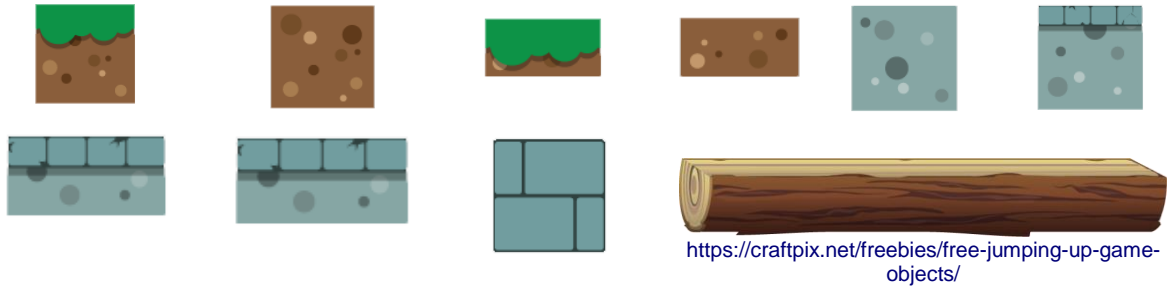
Para el desarrollo del código fuente del videojuego (scripts).

4.3 Assets y recursos del juego

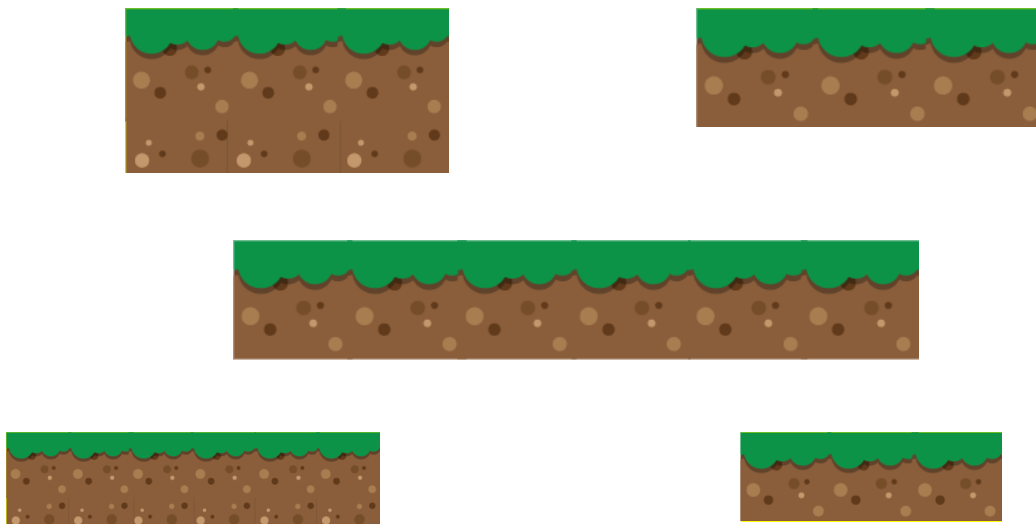
Suelo y plataformas

(<http://www.imaginelabs.rocks/?product=platform-tile-set>)

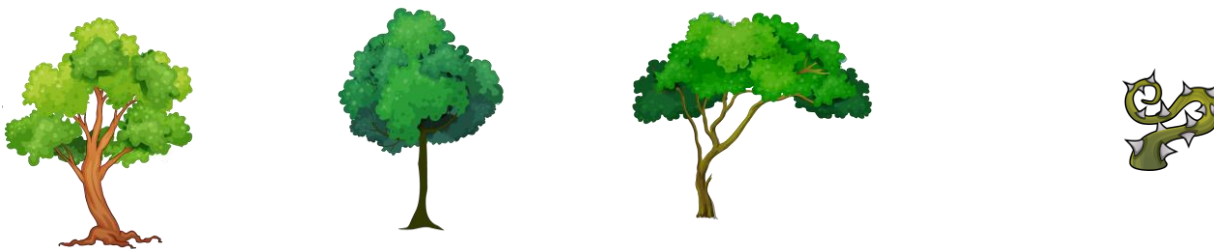
A partir de las siguientes pequeñas piezas he ido montando el escenario manualmente. Estos assets son del enlace indicado, a excepción del tronco.



Con Photoshop he creado las siguientes superficies, más grandes que las unidades anteriores, para rellenar grandes espacios del nivel.



Vegetación



https://favpng.com/png_view/tree-tree-planting-royalty-free-illustration-png/urCEM3jw/

<https://www.hiclipart.com/free-transparent-background-png-clipart-fginn/download>



<https://craftpix.net/freebies/free-jumping-up-game-objects/>



<https://www.pngfly.com/png-mf3rcx/download.html>



http://www.imaginelabs.rocks/?page_id=168

Medallas



<https://www.hiclipart.com/free-transparent-background-png-clipart-bfwa>

Otros elementos



<https://www.101computing.net/pygame-how-to-control-your-sprite/>



<https://craftpix.net/freebies/free-jumping-up-game-objects/>



http://www.imaginelabs.rocks/?page_id=168



http://www.imaginelabs.rocks/?page_id=168



<https://owips.com/clipart-15409434>



<http://unluckystudio.com/free-game-artassets-for-games-11-ultimate-platformer-game-art-pack/>



Background

El fondo del videojuego es el siguiente:

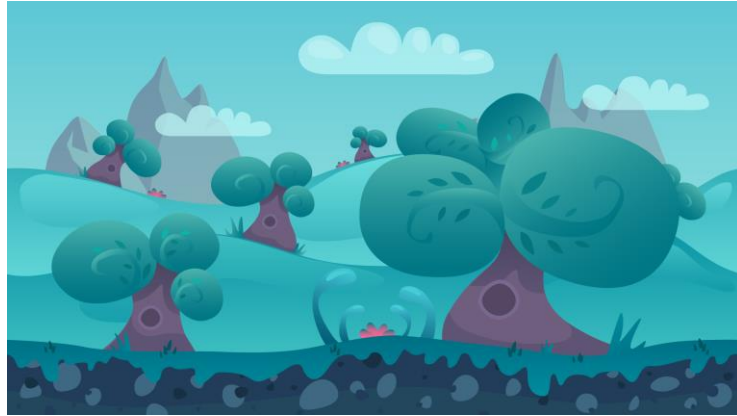


Ilustración 17: Fondo del videojuego
<http://unluckystudio.com/11-free-game-backgrounds/>

Consta de siete capas independientes que he programado para que se muevan a distintas velocidades (parallax).

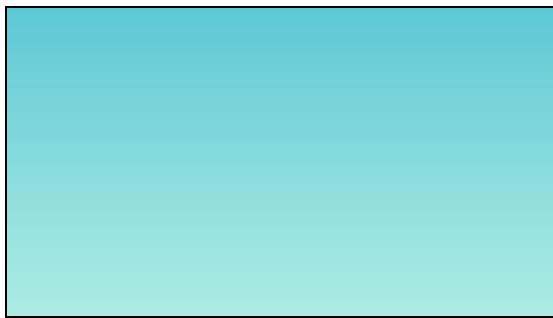


Ilustración 18: Capa 7 - cielo

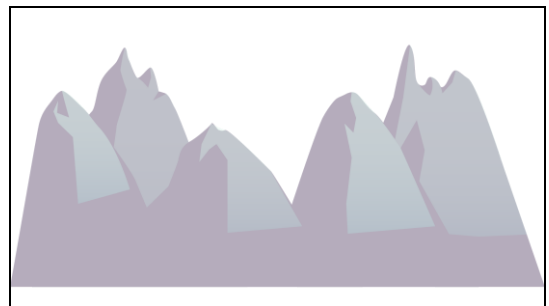


Ilustración 19: Capa 6 - montañas



Ilustración 20: Capa 5 - nubes

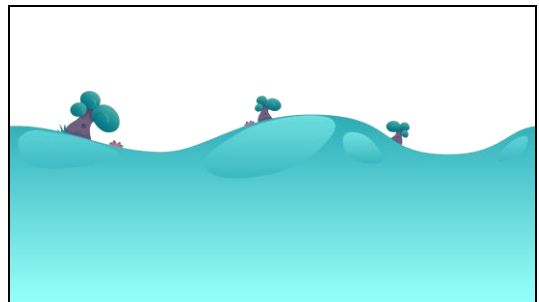


Ilustración 21: Capa 4 - colinas3

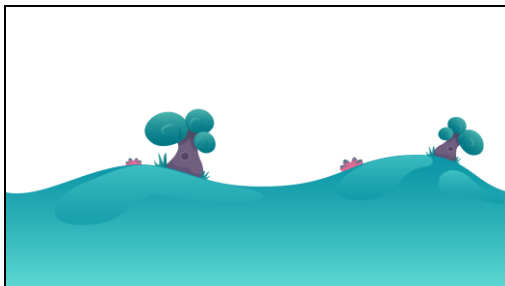


Ilustración 22: Capa 3 - colinas2

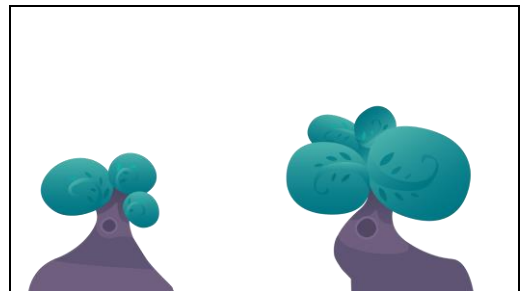


Ilustración 23: Capa 2 - colinas 1

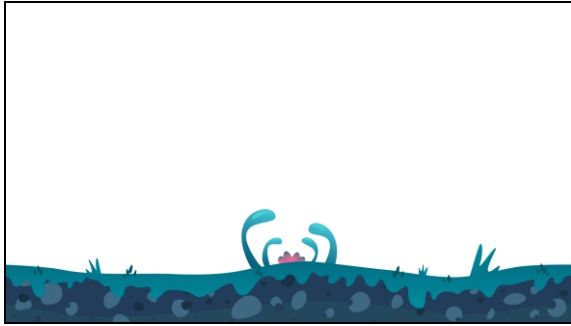


Ilustración 24: Capa 1 - suelo

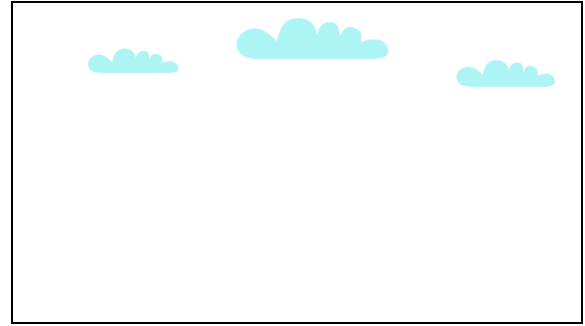


Ilustración 25: Capa nueva del cielo

La capa nueva del cielo sustituye a la original porque ésta tenía un poco de transparencia, lo que hacía que se vieran las montañas. Lo que he hecho ha sido quitarle la transparencia y, también, subir un poco la nube que está a la izquierda para que encaje mejor en la composición final.

Personaje principal

El personaje principal lo he comprado de la página Dreamstime.com (<https://www.dreamstime.com/print-image146622236>)

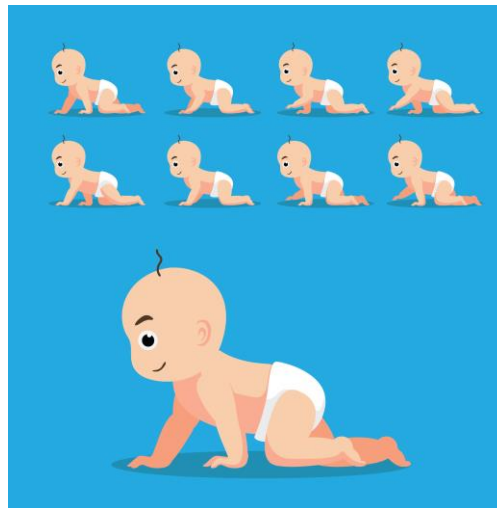


Ilustración 26: Personaje principal del juego

ID146622236 © Pun nawich Limparungpatanakij | Dreamstime.com

A partir de este asset y de su sprite de ganeo he ido trabajando con Photoshop para obtener la figura deseada.

En primer lugar, he borrado su boca y he dibujado un chupete. Después he girado la imagen para poder tener los sprites de ganeo en ambos sentidos.

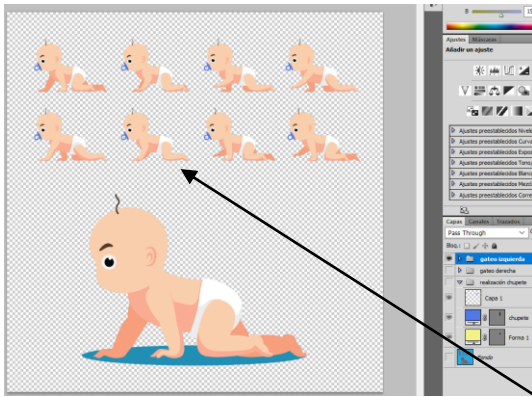


Ilustración 27: sprite gateo izquierda

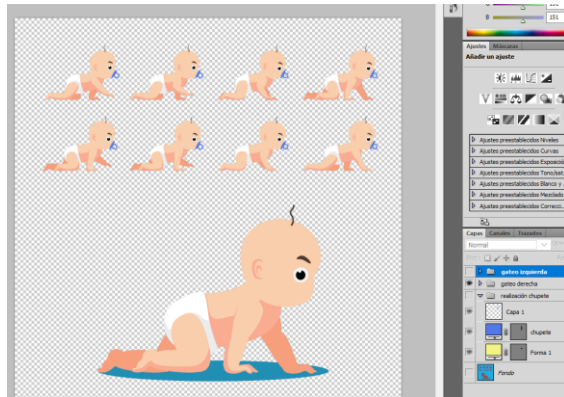


Ilustración 28: sprite gateo derecha

He aprovechado una de las figuras del gateo para multiplicarla y hacer la figura idle del correspondiente Sprite (reposo). Aquí he querido darle movimiento al chupete, de manera que en cada figura está más cerca o alejado del rostro. De esta forma en el videojuego se puede apreciar dicho movimiento para darle un toque más divertido.

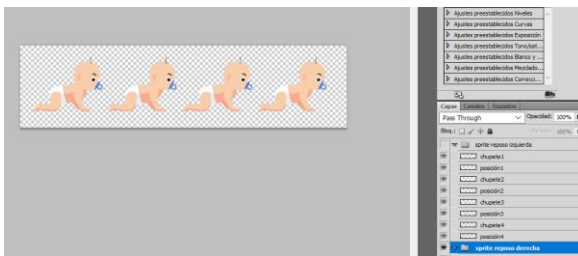


Ilustración 29: idle derecha

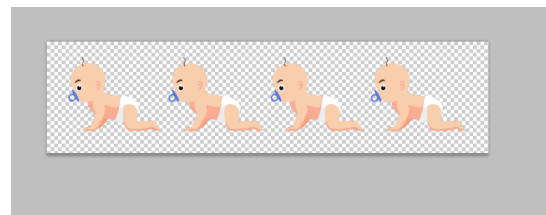


Ilustración 30: idle izquierda

Por último, he trabajado el movimiento de las extremidades del personaje para poder hacer el Sprite del salto.

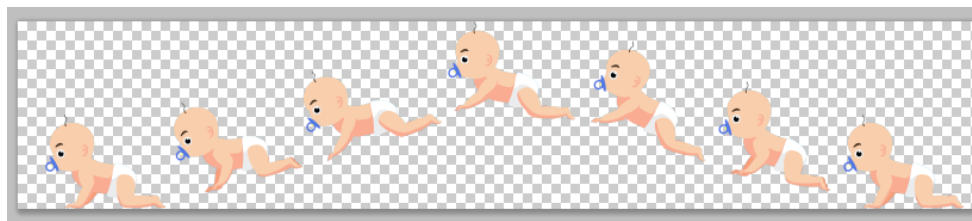
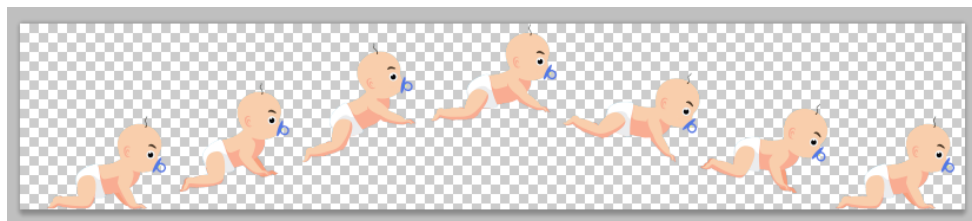


Ilustración 31: saltos derecha e izquierda

Enemigos

Los tres enemigos que aparecen en el videojuego.



Ilustración 32: enemigos del videojuego
<https://craftpix.net/freebies/>

Biberón

Este asset ha sido creado por mí. Después he hecho varias copias del biberón para poder crear el sprite.

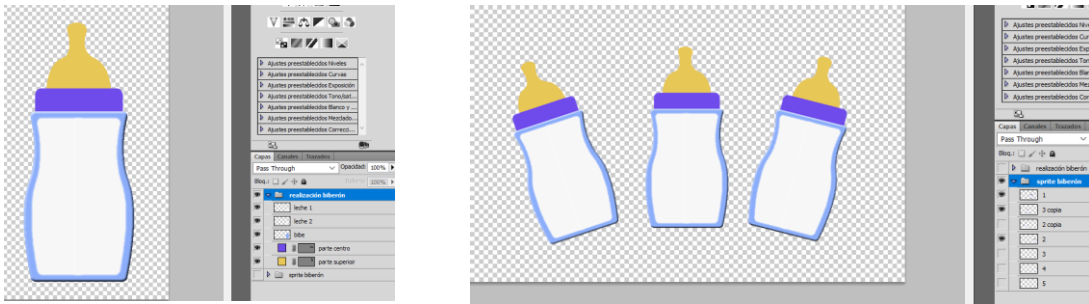


Ilustración 33: creación y sprite del coleccionable

RECURSOS SONOROS

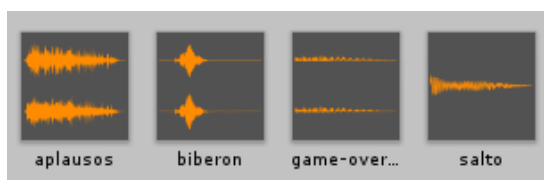


Ilustración 34: pistas de audio para los efectos sonoros
<https://freesound.org/>

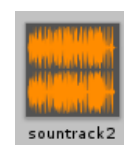
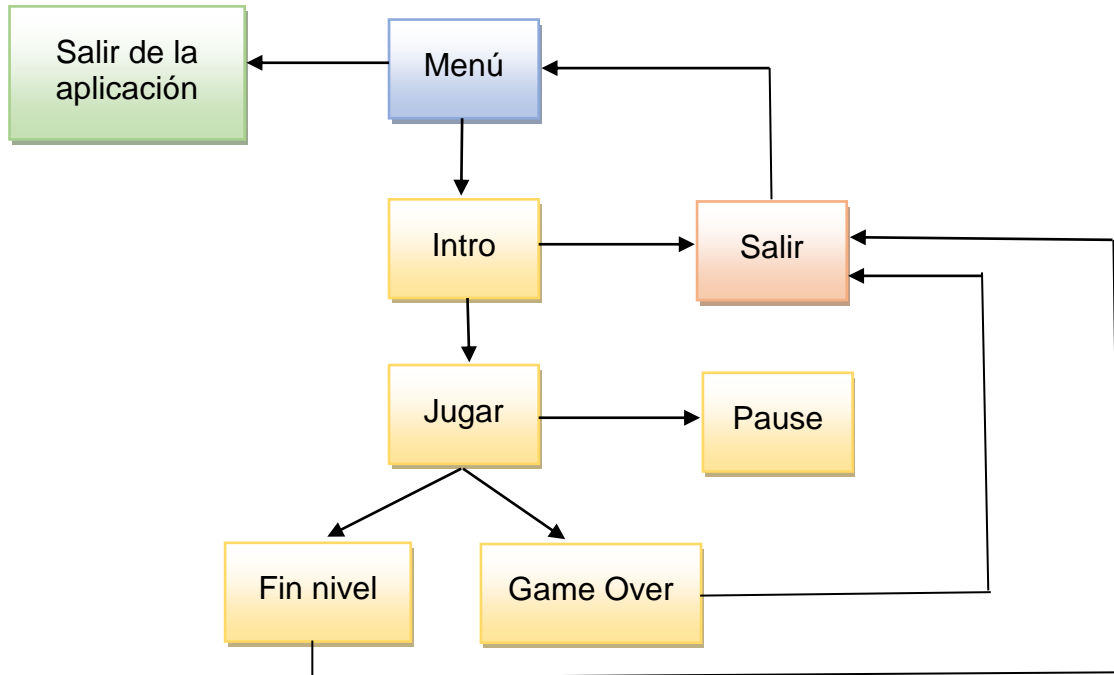


Ilustración 35: música de fondo del juego
<https://www.bensound.com/royalty-free-music>

4.4 Esquema de arquitectura del juego

El esquema del juego es el siguiente:



Las dos clases más importantes del videojuego son el Game Manager y el jugador.

- Controlador de la IU es el Game Manager: GameManager.cs
- El controlador del jugador es PlayerController.cs

4.5 IA de los enemigos

La IA de los enemigos es básica. He utilizado las animaciones: *idle*, *walking* y *running*. En el caso de los enemigos que caminan y corren he programado un target para que recorran una distancia hasta dicho target y vuelvan al punto de inicio.



Ilustración 36: Idle del enemigo 1

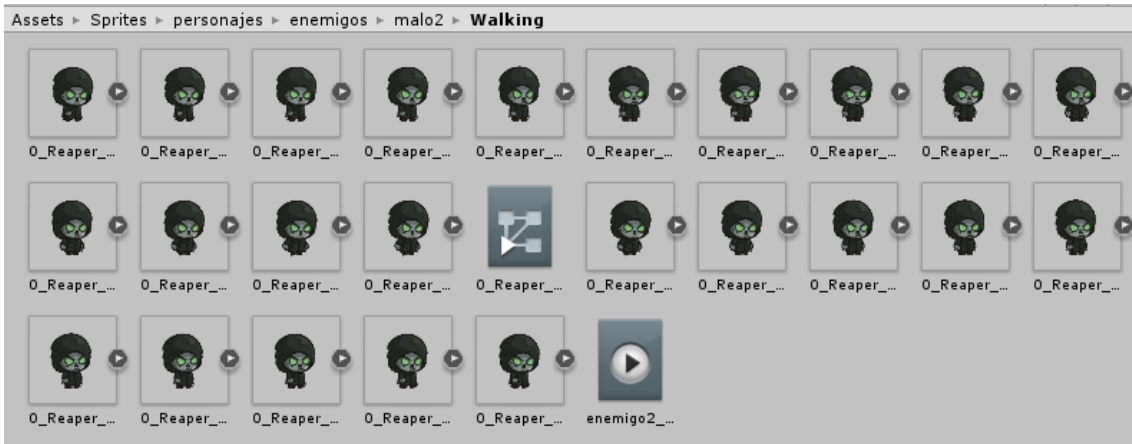


Ilustración 37: Walking enemigo 2

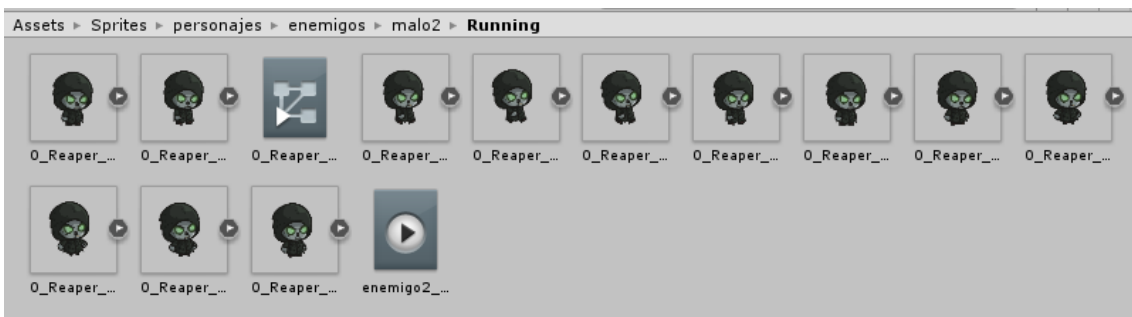


Ilustración 38: Running enemigo 2

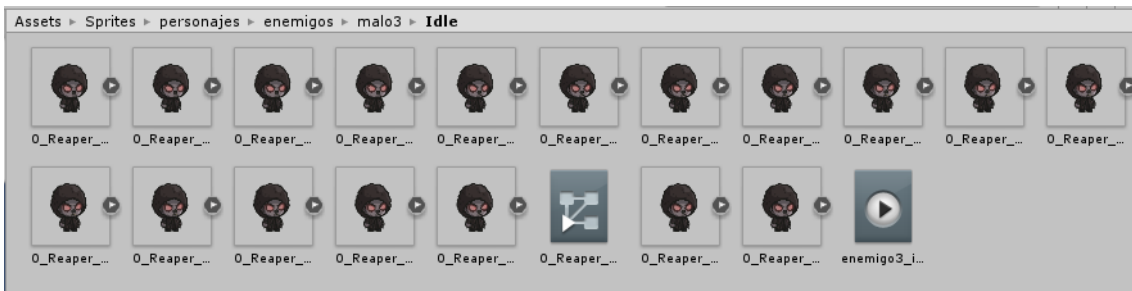


Ilustración 39: Idle enemigo 3

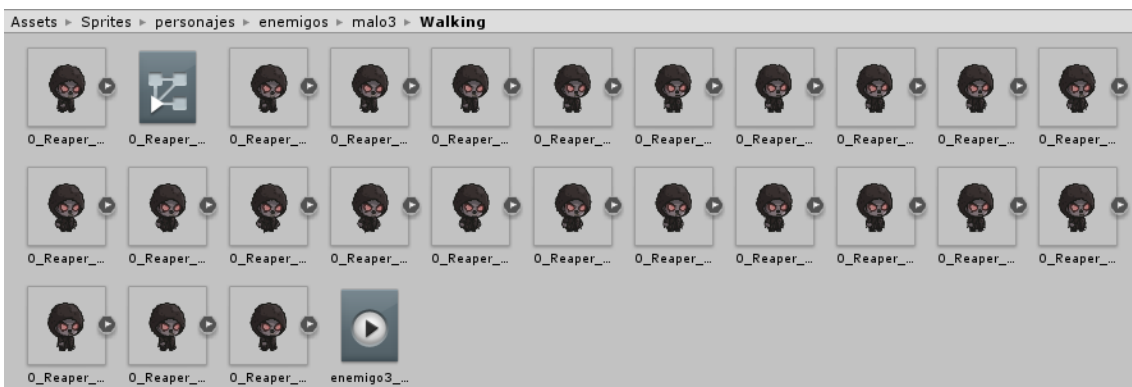


Ilustración 40: Walking enemigo 3

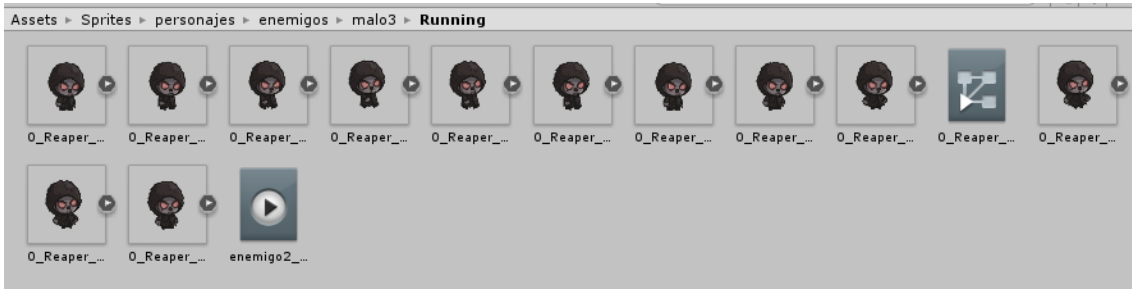


Ilustración 41: Running enemigo 3

5. Diseño de niveles

El mapa del nivel es el siguiente:

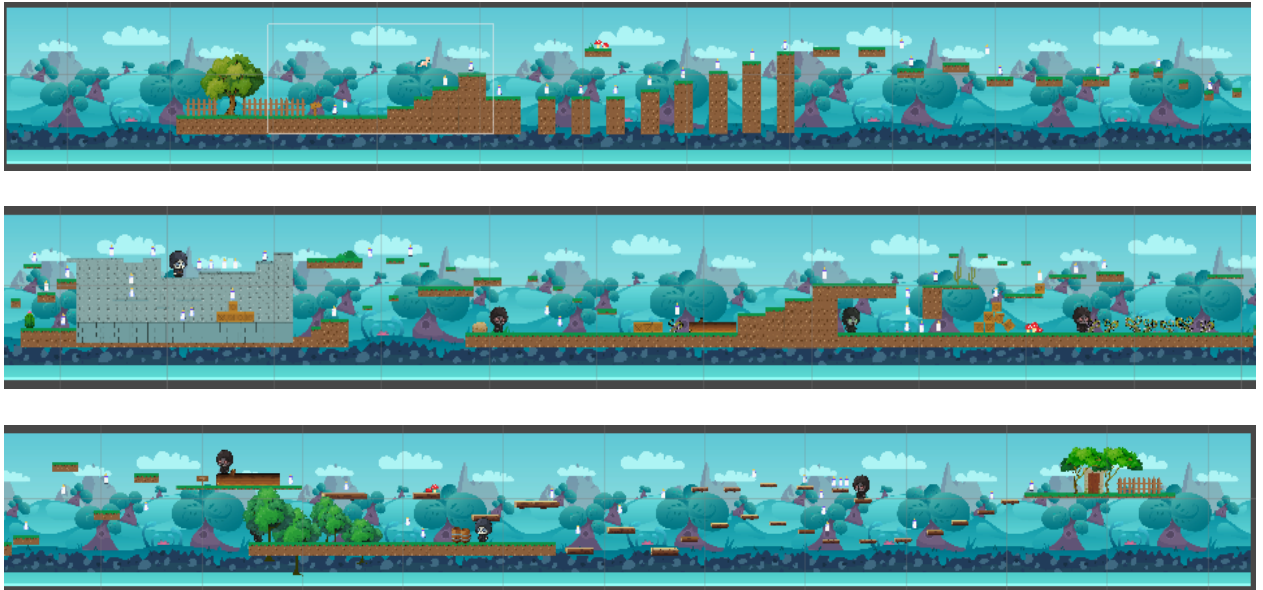


Ilustración 42: Mapa del nivel del juego

He creado un único nivel. Para ello, primero he empezado por un escenario pequeño y, poco a poco, he ido aumentándolo teniendo en cuenta la ubicación de las plataformas para que el personaje pueda llegar a ellas cuando salta. Por otro lado, he tenido en cuenta que el bebé puede retroceder en cualquier momento para recoger biberones, por lo que he diseñado un camino de vuelta para que ello sea posible.

El jugador tiene que recoger un mínimo de 40 biberones para poder superar el nivel. A partir de ahí, y en función de los que haya recogido al final del escenario, obtendrá una medalla. Con 40 ganará la de bronce, a partir de 55 la de plata y si consigue superar la cifra de 70 biberones, entonces ganará la de oro.

El juego no presenta una gran dificultad porque se trata de un bebé que gatea, pero aún así he tratado de poner algunos enemigos en zonas concretas para que el jugador tenga que esperar antes de poder seguir avanzando.

6. Manual de usuario

6.1 Requerimientos técnicos del hardware

Los requisitos mínimos son:

- **Sistema operativo:** Windows 7 / 8 / 10
- **Procesador:** Core 2 Duo o superior
- **Memoria:** 1 GB de RAM
- **Gráficos:** DirectX11 Compatible GPU con 512 MB Video RAM
- **Almacenamiento:** 100 MB de espacio disponible
- **Tarjeta de sonido:** DirectX compatible Tarjeta de sonido

6.2 Instrucciones del juego

En la siguiente imagen se indican cuáles son los controles y, también, cuál es el objetivo del juego.



Ilustración 43: Instrucciones del juego

7. Conclusiones

Conclusiones del trabajo

En primer lugar, he aprendido que un trabajo de estas características necesita una buena planificación para poder lograr los objetivos planteados en el inicio. Aún así siempre surgen imprevistos, por lo que es muy importante tenerlo en cuenta a la hora de realizar dicha planificación del proyecto.

Por otro lado, me ha parecido muy interesante realizar este trabajo porque se ha tratado de mi primer contacto con el mundo del desarrollo de videojuegos. Hace años habría pensado que era imposible que pudiera hacerlo, pero hoy puedo decir que siento una gran satisfacción personal.

Logro de los objetivos planteados

He logrado casi todos los objetivos, pero alguno se ha quedado en el tintero a pesar de estar planificado. Tenía la intención de hacer pruebas con usuarios, pero la falta de tiempo no me lo ha permitido. Esto ha sido así porque he necesitado aprender a manejar Unity, ya que ha sido una herramienta nueva para mí.

Seguimiento de la planificación y metodología

He seguido la planificación, pero si tuviera la oportunidad de empezar de nuevo la haría diferente. Al tratarse de mi primer videojuego no tenía la menor idea de todo lo que ello conllevaba. La metodología prevista ha sido adecuada, pero no tanto la planificación, ya que algunos errores de programación me han retrasado.

Algunos cambios que he introducido en el trabajo para mejorar el nivel han sido:

- Crear la pantalla **Intro** para que el jugador no empezara a jugar nada más arrancar el juego.
- Crear la pantalla **Pause**. No la había tenido en cuenta y es de vital importancia.
- Poner un objetivo en la recogida de biberones para que el jugador pueda ganar una medalla al final de la partida.
- Añadir más efectos sonoros.

Me habría gustado incorporar otros cambios sugeridos por el consultor, pero no lo he conseguido. Estos cambios y/o arreglos son:

- Hacer que algunas plataformas puedan para que el jugador no se entretenga y pase rápido.
- Deshabilitar el botón **Jugar** del menú inicial para que el jugador se vea obligado pulsar sobre **Intro**.

Líneas de trabajo futuro

Con los conocimientos adquiridos durante este semestre, de cara al futuro lo que haría sería:

- Añadir los movimientos de subir/bajar del personaje (no lo he logrado hacer ahora).
- Modificar el nivel para hacerlo más difícil.
- Programar el producto para que pueda exportarse a plataformas móviles.
- Incorporar realidad aumentada.

8. Glosario

Asset: cualquier elemento que forma parte de un proyecto.

Sprite: imagen de mapa de bits presente en la pantalla, generalmente personajes de videojuegos.

IU: Interfaz de usuario

IA: Inteligencia artificial

9. Bibliografía

- <https://www.youtube.com/watch?v=wjd7pifLpAc>
- <https://histinf.blogs.upv.es/2012/12/14/historia-de-las-consolas/>
- <https://docs.hektorprofe.net/escueladevideojuegos/academia-unity/#juego-de-plataformas-2d>
- <https://www.youtube.com/watch?v=wdfDW3PMZQI>
- <https://assetstore.unity.com/packages/2d/environments/free-platform-game-assets-85838>
- <https://www.gamedevmarket.net/category/2d/backgrounds/>
- <https://frogames.es/recursos-arte-gratuitos-desarrolladores-videojuegos-indie/>
- <https://rgoncalves83.itch.io/free-vector-grass-tileset>
- <https://www.gameart2d.com/free-platformer-game-tileset.html>
- <https://vandal.elespanol.com/requisitos/pc/unity-of-four-elements/56854#p-13>
- <https://articulo.mercadolibre.com.ar/MLA-820044071-cartucho-super-mario-bros-1-2-3-p-family-game-nes- JM>
- <https://www.delabahia.com.ar/se-cumplen-38-anos-del-lanzamiento-del-juego-donkey-kong/>
- <https://store.unity.com/es/compare-plans>
- [1] https://es.wikipedia.org/wiki/GameMaker_Studio
- https://commons.wikimedia.org/wiki/File:GameMaker_Studio_2_logo.png
- <https://www.aprendegamemaker.com/game-maker-2/>
- <https://www.yoyogames.com/>
- <https://gdevelop-app.com/>
- https://fr.wikipedia.org/wiki/Fichier:Logo_GDevelop5.png
- <https://codelearn.es/cursos/curso-de-videojuegos-2d-con-unity/>

- <https://docs.unity3d.com/Manual/system-requirements.html>
- [2]<https://www.istockphoto.com/mx/vector/hoja-de-sprite-de-animaci%C3%B3n-rastreo-beb%C3%A9-gm917310982-252368832>
- <http://www.gamerdic.es/termino/sprite/>