

MY FIRST HERO



David Reyes Tejada

Grado de Ingeniería Informática

Videojuegos

Consultor: *Jordi Duch Gavaldá, Heliodoro Tejedor Navarro*

Profesor: *Joan Arnedo Moreno*

6 de Enero de 2019

COPYRIGHT

Esta obra está sujeta a una licencia de
Reconocimiento-NoComercial-
SinObraDerivada



© David Reyes Tejada

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

DEDICATORIA

“A mi hijo. Para que, pase lo que pase, nunca piense que no puede conseguir lo que se proponga.

Las barreras solo las ponemos nosotros y de cada caída nos levantamos aún mas fuertes”

“A mis padres. Por traerme hasta aquí y acercarme al mundo que me apasiona y, por no perder nunca la esperanza”

“A mi compañera, guía y mitad. Siempre ha estado ahí para ayudar y sin ella, nunca habría conseguido esto”

AGRADECIMIENTOS

A todos los que me habéis escuchado,
aconsejado y orientado en este camino

Muchas Gracias a todos
por recorrerlo conmigo

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>TFG Videojuegos – My First Hero</i>
Nombre del autor:	<i>David Reyes Tejada</i>
Nombre del consultor/a:	<i>Jordi Duch Gavaldá</i> <i>Heliodoro Tejedor Navarro</i>
Nombre del Profesor:	<i>Joan Arnedo Moreno</i>
Fecha de entrega (mm/aaaa):	01/2019
Titulación:	<i>Grado de Ingeniería Informática</i>
Área del Trabajo Final:	<i>Videojuegos</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Videojuego, Aventura, GameMaker</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i>	
<p>El objetivo de este proyecto es el desarrollo e implementación de un videojuego como parte del trabajo de fin de grado para los estudios de informática. Usaremos los conocimientos adquiridos durante nuestro estudio y los aplicaremos a una serie de herramientas para la consecución del trabajo.</p> <p>El título del videojuego es “<i>My First Hero</i>”, un juego de aventuras/acción que intentará acercar a los jugadores más inexpertos o jóvenes, el mundo del videojuego de aventuras, donde la historia y la relación con otros personajes, es tan importante como la propia acción. Todo ello mediante un enfoque sencillo y unas mecánicas directas.</p>	

El mayor reto al que nos enfrentábamos era el uso de la propia herramienta de desarrollo, "GameMaker". Este software era totalmente desconocido, así como sus funcionalidades, por lo que hemos seguido una metodología de aprendizaje y aplicación continua, y dados los resultados, podemos afirmar que se han implementado la mayoría de las ideas propuestas.

Podemos concluir con una gran satisfacción por el trabajo realizado y los conocimientos adquiridos. La implementación de muchas de las ideas ha requerido de largas sesiones de aprendizaje que se han visto reflejadas positivamente. El proyecto aún tiene un gran margen de mejora y podrá ser expandido en los próximos meses.

Abstract (in English, 250 words or less):

The objective of this project is the development and implementation of a videogame as part of the closure project to baachelor's degree in computer studies. We will use the knowledge acquired during our study and apply it to a series of tools to achieve the job.

The title of the videogame is "My First Hero", an adventure / action game that will try to bring the most inexperienced or young players, the world of adventure videogames, where the story and the relationship with other characters, is as important as the action itself. All this through a simple approach and direct mechanics.

The biggest challenge we faced was the use of the development tool itself, "GameMaker". This software was totally unknown, as well as its functionalities, so we have followed a methodology of learning and continuous application, and given the results, we can affirm that most of the proposed ideas have been implemented.

We can conclude with great satisfaction for the work done and the knowledge acquired. The implementation of many of the ideas has required long learning sessions that have been positively reflected. The project still has a great margin for improvement and may be expanded in the coming months.

INDICE

1. INTRODUCCIÓN	1
1.1 CONTEXTO Y JUSTIFICACIÓN DEL TRABAJO	1
1.2 OBJETIVOS DEL TRABAJO	2
1.3 ENFOQUE Y MÉTODO SEGUIDO	3
1.4 PLANIFICACIÓN DEL TRABAJO	5
1.5 BREVE SUMARIO DE PRODUCTOS OBTENIDOS	6
1.6 BREVE DESCRIPCIÓN DE LOS OTROS CAPÍTULOS DE LA MEMORIA	6
2. ESTADO DEL ARTE	7
2.1 GENERO DEL JUEGO	7
2.1.1 EVOLUCIÓN DEL REFERENTE	8
2.2 PLATAFORMAS UTILIZADAS PARA EL DESARROLLO	14
2.2.1 GAMEMAKER STUDIO 2	15
2.2.2 UNREAL ENGINE	16
2.2.3 UNITY	17
2.2.4 CONSTRUCT3	17
2.3 VALORACIÓN ECONÓMICA DEL TRABAJO	18
2.3.1 SOFTWARE	18
2.3.2 HARDWARE	18
2.3.3 RECURSOS HUMANOS	18
2.4 PLATAFORMA DE DESTINO	20
3. DEFINICIÓN DEL JUEGO	21
3.1 HISTORIA Y AMBIENTACIÓN	21
3.1.1 DESARROLLO DEL PRIMER NIVEL	21
3.2 OBJETIVOS PLANTEADOS AL JUGADOR	22
3.3 PERSONAJES	23
3.3.1 PERSONAJES Y ELEMENTOS DE INTERACCIÓN CON EL HÉROE	23
3.4 CONCEPT ART	24
3.4.1 EL HÉROE	24
4. DISEÑO TÉCNICO	28
4.1 ENTORNO ESCOGIDO Y REQUISITOS	28
4.2 HERRAMIENTAS UTILIZADAS	30
4.2.1 ADOBE PHOTOSHOP CS6	30

4.2.2 ADOBE PREMIERE PRO	32
4.3 INVENTARIO DE ASSETS Y RECURSOS DEL JUEGO	32
4.3.1 SPRITES	33
4.3.2 SONIDO	34
4.4 ARQUITECTURA DEL JUEGO DISEÑO	35
4.4.1 ANIMACIÓN Y DISEÑO DE PERSONAJES	35
4.4.2 LÓGICA DEL PERSONAJE	37
4.4.3 OBJETOS INTERACTIVOS Y COLISIONES	49
4.4.4 ENEMIGOS E IA ENEMIGA	50
4.4.5 OBJETOS DESTRUCTIBLES Y ANIMALES	51
4.4.6 SONIDO	52
4.4.7 HUD	52
5. DISEÑO DE NIVELES	53
5.1 PROCESO DE DISEÑO	53
5.1.1 SPRITES	55
5.1.2 CONJUNTO DE CASILLAS	58
5.1.3 DISEÑO DE MAPAS	58
5.1.4 ELEMENTOS NECESARIOS DEL DISEÑO DE NIVELES	62
6. MANUAL DE USUARIO	71
6.1 REQUISITOS DEL JUEGO	71
6.2 INSTRUCCIONES DEL JUEGO	72
7. CONCLUSIONES	73
7.1 CONCLUSIONES DEL TRABAJO	73
7.2 OBJETIVOS PLANTEADOS. REFLEXIÓN CRÍTICA	74
7.3 SEGUIMIENTO DE LA PLANIFICACIÓN Y METODOLOGÍA	75
7.4 LÍNEAS DE TRABAJO FUTURO	76
8. GLOSARIO	77
9. BIBLIOGRAFÍA	78
10. ANEXOS	¡ERROR! MARCADOR NO DEFINIDO.

Tabla de Ilustraciones

Ilustración 1. Diagrama de Gantt de la planificación	5
Ilustración 2. GameMaker	15
Ilustración 3. Unreal Engine	16
Ilustración 4. Unity Engine.....	17
Ilustración 5. Construct3 Engine.....	17
Ilustración 6. Tabla de costes en recursos humanos	19
Ilustración 7. Arte del protagonista	24
Ilustración 8. Arte del protagonista.....	25
Ilustración 9. Arte del Jefe de la hermandad	26
Ilustración 10. Arte de enemigos	27
Ilustración 11. Plataforma escogida	28
Ilustración 12. Requisitos plataforma	29
Ilustración 13. Adobe Photosthop Logo.....	30
Ilustración 14. Diseño en Photoshop.....	31
Ilustración 15. Adobe Premiere Pro.....	32
Ilustración 16. Humble Bundle RPG Pack.....	32
Ilustración 17. Lista de sprites	33
Ilustración 18. Lista de sprites	33
Ilustración 19. Lista de sprites	33
Ilustración 20. Lista de sonidos incluidos	34
Ilustración 21. Animación del héroe	35
Ilustración 22. Animación del guardia real.....	35
Ilustración 23. Conversación anciano HUD	36
Ilustración 24. Conversación chico HUD	36
Ilustración 25. Conversación guardia real HUD.....	36
Ilustración 26. Estados Héroe	37
Ilustración 27. Sprite parado héroe	46
Ilustración 28. Sprite andando héroe.....	47
Ilustración 29. Sprite muriendo héroe.....	47
Ilustración 30. Sprite atacando héroe.....	48
Ilustración 31. Objeto interactivo puerta	49
Ilustración 32. Objeto interactivo cartel	49
Ilustración 33. Estados posibles de los enemigos	50
Ilustración 34. Objeto destructible animal.....	51
Ilustración 35. Objeto destructible planta	51
Ilustración 36. HUD	52
Ilustración 37. Puntuación	52
Ilustración 38. Estado inicial de una room	53
Ilustración 39. Configuración de tamaño de una room	54
Ilustración 40. Sprite de diseño de bosque	55
Ilustración 41. Sprite de diseño de ciudad.....	56
Ilustración 42. Sprite de diseño de interior	56

Ilustración 43. Sprite de diseño de mina	57
Ilustración 44. Conjunto de sprites de mapa completo	57
Ilustración 45. Definición del conjunto de casillas	58
Ilustración 46. Capas del diseño del mapa Ciudad	58
Ilustración 47. Mapa Cascada	59
Ilustración 48. Mapa Campo de Damaris	59
Ilustración 49. Mapa Ciudad de Damaris	60
Ilustración 50. Mapa Taberna de Damaris	60
Ilustración 51. Mapa Hermandad de aventureros.....	61
Ilustración 52. Mapas de la mina.....	61
Ilustración 53. Objeto interactivo cartel 1	62
Ilustración 54. Objeto interactivo cartel 2	62
Ilustración 55. Objeto interactivo cofre 2	63
Ilustración 56. Objeto interactivo cofre 1	63
Ilustración 57. Objeto interactivo puerta 2	63
Ilustración 58. Objeto interactivo puerta 1	63
Ilustración 59. Objeto interactivo moneda 2	63
Ilustración 60. Objeto interactivo moneda 1	63
Ilustración 61. Objeto interactivo corazón 1	64
Ilustración 62. Objeto interactivo corazón 2	64
Ilustración 63. Objeto interactivo flores 1	64
Ilustración 64. Objeto interactivo flores 2	64
Ilustración 65. Objeto interactivo animales 2.....	64
Ilustración 66. Objeto interactivo animales 1	64
Ilustración 67. Objeto interactivo NPC's	65
Ilustración 68. Objeto interactivo NPC Rey	65
Ilustración 69. Enemigo soldado esqueleto.....	66
Ilustración 70. Enemigo guerrero esqueleto.....	66
Ilustración 71. Colisiones en el mapa de la mina	67
Ilustración 72. Diseño completo del mapa Campo de Damaris.....	68
Ilustración 73. Lógica de diseño entre zonas	70
Ilustración 74. Controles del juego	72

1. Introducción

1.1 Contexto y justificación del Trabajo

Las nuevas tecnologías no han surgido de la nada. A pesar de que los videojuegos son un fenómeno relativamente nuevo, su origen tuvo una relación muy cercana con la industria del entretenimiento y, desde su aparición, no ha parado de crecer hasta convertirse en lo que es a día de hoy: un reclamo de masas.

Es difícil señalar cual fue el primer videojuego, pero se considera como tal el Nought and crosses, también llamado OXO, que fue desarrollado por Alexander S. Douglas en 1952. OXO es una versión electrónica del clásico juego del tres en raya cuyo objetivo era mostrar la interacción entre el ser humano y un ordenador y se ejecutaba sobre la EDSAC.¹ En este caso, no existía un oponente, sino que el jugador era quien se enfrentaba a la máquina.

El crecimiento de los videojuegos empieza cuando aparece Pong, una máquina recreativa, similar al Tenis for Two, pero que podíamos verla en lugares públicos. El sistema fue diseñado por Al Alcorn para Nolan Bushnell en Atari.

Llamamos videojuego a un juego electrónico (software) creado para el entretener mediante la interacción producida entre varias personas o entre una persona y el dispositivo electrónico. Dicho dispositivo puede ser un ordenador, una videoconsola, una máquina arcade o una plataforma (un teléfono móvil, por ejemplo)

Dentro del mundo de los videojuegos, los hay sencillos y otros mucho más complejos que se pueden considerar verdaderas obras de arte. El objetivo de todos: impulsar las habilidades de las personas. *“Jugar con videojuegos implica*

poner en marcha muchas de nuestras capacidades y habilidades, necesitamos concentración, atención, control, y mucha, pero mucha emoción” (Gil, A.Vida, T.2007:33-34) Es por eso que jugar a los videojuegos en este momento, permite a las personas socializarse en el mundo de las TIC.

De hecho, es gracias a los videojuegos que las personas pueden aprender con mayor facilidad el dominio de las nuevas tecnologías, adquirir capacidades nuevas y desarrollar diversas habilidades, dejando de considerar al videojuego un elemento de aislamiento y sí, un elemento formativo y social, ya que se puede jugar en casa o cibercafés compartiendo y entendiendo esos lugares como una modalidad diferente de socializar y donde se tiene la oportunidad de expresar emociones que son reales en un contexto virtual.

Nuestro proyecto intenta incluir a todos aquellos jugadores inexpertos o que se inician desde un planteamiento de mecánicas simple, un estilo visual llamativo y una historia sencilla. Queremos que el jugador consiga el objetivo principal de jugar a un videojuego, divertirse.

1.2 Objetivos del Trabajo

A la hora de realizar un proyecto, es necesario establecer unos objetivos, principales y secundarios, para conseguir realizarlo con éxito y plantear una serie de fases para buscar la manera más óptima de desarrollar el proyecto: una primera fase para analizar las tecnologías necesarias para llevarlo a cabo, la segunda fase para desarrollarlo y conseguir un producto que se pueda jugar, y la última para realizar un documento del proceso (memoria).

- **Objetivos principales:**
 - Desarrollar un videojuego 2D para Windows con GameMaker que cumpla como objetivo entretener al jugador.
 - Crear un producto con una estética y aspecto original y llamativo.

- Crear, animar e implementar mecánicas para los personajes del videojuego.
- Crear varios escenarios con diferentes recursos visuales para completar el nivel en un tiempo aproximado de 5 minutos
- Comprobar la funcionalidad
- **Objetivos secundarios:**
 - Aprender el manejo del programa de edición GameMaker
 - Usar los conocimientos de Programación aplicándolo al diseño de los videojuegos.

1.3 Enfoque y método seguido

Detrás del desarrollo de un videojuego hay diferentes perfiles profesionales necesarios para llevarlo a cabo.

- **Game Designer:** persona que desarrolla el documento, es decir, define como será el juego
- **Programador:** plasma en código los conceptos que previamente a marcado el Game Designer.
- **Artista:** diseña todo lo que se verá en el título, crea los menús y los sistemas de navegación del juego. Distinguimos entre 2D y 3D
- **Animador:** encargado de crear el “esqueleto” del personaje y después animarlo.
- **Diseñador de audio:** persona encargada de crear la música y efectos de sonido. Trabajo esencial en cualquier juego.

Para la organización de todo el equipo se utilizan diversas metodologías para hacer más fácil la planificación del trabajo. En este caso, como no vamos a desarrollar el proyecto con un equipo multidisciplinar, no necesitamos ningún tipo de metodología estricta.

Es por ello que desde el inicio del proyecto ha sido necesario combinar la parte artística con la parte de desarrollo de programación. En un primer acercamiento a la implementación hemos diseñado una versión base sobre la que dar luz a las mecánicas que debería tener nuestro proyecto.

En este punto es necesario añadir, que el diseño original junto con todos los sprites de arte fue desechado por no llegar a los mínimos de calidad y mecánicas que nos auto exigíamos.

Cabe decir, que el Feedback recibido por parte de terceras personas ha sido fundamental para realizar los cambios necesarios para adaptar las ideas de forma óptima al proyecto.

1.4 Planificación del Trabajo

Para mostrar la planificación de objetivos de este proyecto se incluye un diagrama de Gantt con los diferentes hitos y fechas junto con la planificación para cada uno de ellos. Esta planificación realizada en los primeros compases del proyecto ha recibido modificaciones debido a los cambios o problemas surgidos durante el desarrollo.

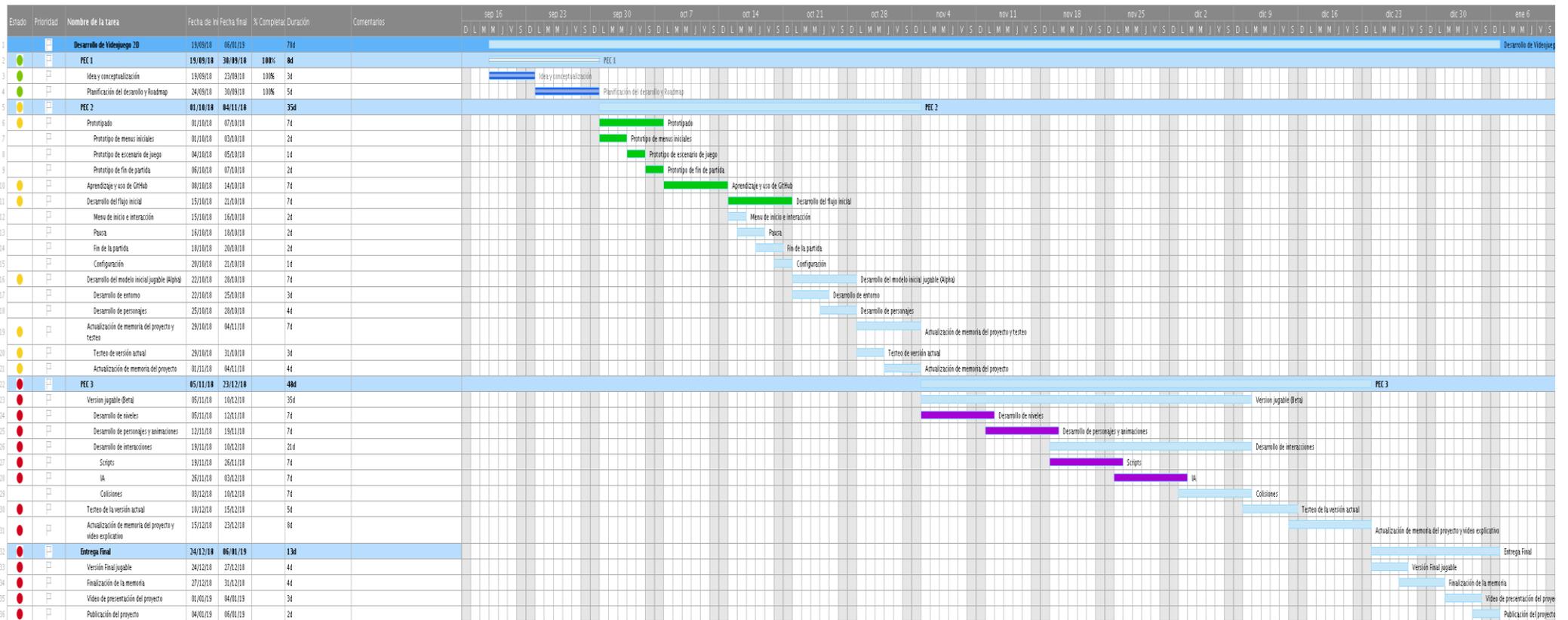


Ilustración 1. Diagrama de Gantt de la planificación

1.5 Breve resumen de productos obtenidos

Durante los meses que ha durado este proyecto hemos obtenido una serie de productos, tanto en forma de conocimientos como tangibles. En este punto enumeraremos los productos generados por el desarrollo:

- Proyecto de videojuego en GameMaker
- Memoria final del trabajo
- Informe de autoevaluación
- Video explicativo comentado

1.6 Breve descripción de los otros capítulos de la memoria

En capítulos posteriores realizaremos un análisis detallado de cada uno de sus aspectos clave.

Por un lado, un estudio del género de nuestro videojuego, en el que veremos que ideas son las que marcaron las bases de juegos de este formato y cuáles son sus mayores referentes.

Posteriormente, abordaremos la parte técnica donde además de pormenorizar los softwares utilizados para la implementación, estudiaremos los puntos claves del código de nuestro proyecto. Junto a ello, veremos otros aspectos del diseño tales como los mapas de juego.

Finalmente, llegaremos a las conclusiones extraídas de este proyecto.

Además de estos puntos encontraremos glosario, referencias, etc., comunes a los trabajos de investigación y desarrollo como este, en los que no realizaremos más hincapié hasta llegar a ellos.

2. Estado del arte

2.1 Genero del juego

Las aventuras conversacionales y las gráficas son uno de los géneros más antiguos de videojuegos y los géneros más utilizados en el área de educación. Aplicamos género aventura a juegos en los que el jugador maneja al héroe protagonista de una serie de andanzas.

Los juegos de aventuras tienen cinco características distintivas:

- Narrativa integrada en el juego
- El jugador controla a un personaje
- La mecánica es encontrar solución a puzles
- La interacción básica en el juego es la manipulación de objetos
- El juego incita a explorar el espacio virtual e interactuar y experimentar con las acciones que puedan tener en él.

Estos rasgos no son exclusivos del género de los juegos de aventuras, pero su aparición simultánea es lo que lo distingue del resto de géneros. Por eso mismo, este juego, no puede estar catalogado únicamente como aventura, sino que es una mezcla de aventura y acción pues, como le sucede a The Legend of Zelda, no es un juego solo de aventuras ya que que la mecánica principal no es darles solución a puzles sino combatir contra enemigos.

En conjunto, los rasgos anteriormente descritos, hacen de este género, uno de los más apropiados y utilizados para motivar al aprendizaje. El jugador va aprendiendo a resolver problemas conforme investiga el mapa, examina y manipula objetos e interactúa con otros personajes y objetos disponibles.

Por otra parte, el género de acción se caracteriza porque el jugador debe hacer uso de la velocidad y destreza para el combate y de tiempo de reacción para avanzar en el nivel. Se trata de un género que es muy amplio, y que permite hacer géneros híbridos con casi cualquier otro género, como es este caso concretamente. De este modo, se recalca The Legend of Zelda, porque ha sido el videojuego de acción-aventura desarrollado por Nintendo, que ha servido de prototipo en el que basar el proyecto.

The Legend of Zelda tiene como protagonista al héroe Link, que es un personaje convocado por el destino a derrotar a Ganondorf, poderoso rey y principal antagonista, para rescatar a la princesa Zelda.

2.1.1 Evolución del referente

Como se puede observar en la siguiente evolución del videojuego Zelda, el género que se plantea en este proyecto ha sufrido cambios drásticos propios de la tecnología usada para sus desarrollos. En el juego que se plantea se realizará un acercamiento a los principios de la saga con una mejora gráfica, pero siguiendo una estructura parecida.

The Legend of Zelda (1986)

Desarrollado en 8 bits, a día de hoy tanto los diseños de sus sprites como algunos detalles del inventario siguen manteniéndose, demostrado lo acertado de su diseño inicial.



Captura de Pantalla 1. The Legend of Zelda (1986)

The adventure of Link (1987)



Captura de Pantalla 2. The Adventure of Link (1987)

Este fue un cambio radical en la evolución puesto que paso a un scroll lateral con una clara mejora gráfica en el personaje.

A link to the Past (1991)



Captura de Pantalla 3. A link to the Past (1991)

De vuelta a la vista área, en el que puede ser el referente de su género. Sprites más detallados, así como los mapeados, una autentica obra maestra.

Link's Awakening (1993)



Captura de Pantalla 4. Link's Awakening (1993)

El paso por GameBoy provocó un paso atrás en lo que a nivel visual se refiere. A su favor, introdujo nuevas escenas donde se podía observar el rostro de Link y, mantenía las mecánicas originales intactas.

Ocarina of Time (1998)



Captura de Pantalla 5. Ocarina of Time (1998)

Considerado el mejor de la saga, su salto a Nintendo64 presentaba un nuevo aspecto gráfico y una narrativa separada entre el Link niño y adulto. Una auténtica joya de los videojuegos.

Majora's Mask (2000)



Captura de Pantalla 6. Majora's Mask (2000)

Con un apartado gráfico continuista, el diseño de Majora's Mask se centró en la mejora de la experiencia jugable. Un gran juego donde pudimos observar las primeras transformaciones en el personaje.

The Wind Waker (2002)



Captura de Pantalla 7. The Wind Waker (2002)

Con el cambio gráfico más radical de la saga, esta entrega de GameCube dio el paso hacia el cell shading manteniendo las mecánicas de la anterior generación. Un episodio que no dejó a nadie indiferente.

Four Swords (2002)



Captura de Pantalla 8. Four Swords (2002)

Esta edición de Game Boy Advance iniciaba el formato de la aventura en compañía recurrente en otros Zelda multijugador.

Twilight Princess (2006)



Captura de Pantalla 9. Twilight Princess (2006)

El capítulo más oscuro de la saga, con un aspecto más adulto que nunca y un gesto más severo. Es el Link más realista que se había presentado hasta la fecha. En este capítulo volvían las transformaciones a la saga.

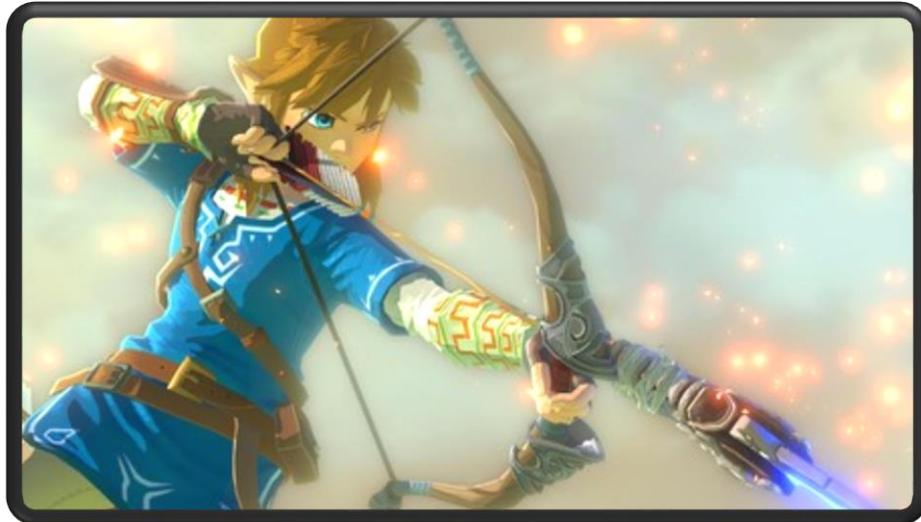
Skyward Sword (2011)



Captura de Pantalla 10. Skyward Sword (2011)

Volviendo a un aspecto más cartoon, esta edición para la consola Wii se centró en el control de movimiento de la consola para adaptar sus mecánicas. Una gran aventura que cosecho una crítica espectacular.

Breath of the Wild (2017)



Captura de Pantalla 11. Breath of the Wild (2017)

El último capítulo de la saga. Una joya atemporal de los videojuegos que presentaba a Link en un mundo abierto con unas posibilidades casi infinitas. Su narrativa, presentada de forma poco convencional, fue elogiada y contrariada, pero es un producto de una calidad inmensa.

Estos son algunos de los capítulos más relevantes de la saga y, apreciando su aspecto gráfico se puede observar como el proyecto gira entorno a los comienzos de la saga adaptándolos en la medida posible a los niveles de calidad y diseño actuales.

2.2 Plataformas utilizadas para el desarrollo

Actualmente el mercado del desarrollo de videojuego cuenta con una serie de Kits o motores de desarrollo bastante profesionales. Estos nos permiten aplicar nuestros conocimientos ayudándonos con herramientas de diseño, por poner un

ejemplo, sería como si a un pintor profesional le damos una aplicación como Photoshop.

En función del Kit y de su versión podemos encontrar gran cantidad de contenido prediseñado como pueden ser personajes, animaciones, objetos de nivel, etc. Es por ello que resulta fundamental escoger el Kit de desarrollo y motor que más se adecue a nuestras necesidades y, sobre todo, a la idea de videojuego que queremos llevar a cabo.

2.2.1 GameMaker Studio 2



Ilustración 2. GameMaker

Es un engine principalmente orientado al 2D, aunque también incluye algunas funcionalidades para el desarrollo 3D. Su creador fue Mark Overmars durante los años 90, y se liberó como herramienta para el desarrollo de videojuegos en el año 1999. El lenguaje de programación que utiliza es propio (GML), bastante similar a Delphi, con una sintaxis sencilla. Uno de los sistemas de creación es de “drag and drop” o coger y soltar, el cual nos permite incluir personajes, enemigos y eventos con solo soltarlos sobre el mapa, puesto que el genera el código necesario para la interacción.

2.2.2 Unreal Engine



Ilustración 3. Unreal Engine

Unreal Engine es un motor de juego de PC y consolas creado por la compañía Epic Games, demostrado inicialmente en el shooter en primera persona Unreal en 1998. Aunque se desarrolló principalmente para los shooters en primera persona, se ha utilizado con éxito en una variedad de otros géneros, incluyendo los videojuegos de sigilo, lucha, MMORPG y otros RPG. Con su código escrito en C++, el Unreal Engine presenta un alto grado de portabilidad y es una herramienta utilizada actualmente por muchos desarrolladores de juegos.

La principal ventaja de este motor es que es gratuito y extremadamente potente. No obstante, aunque fue diseñado para shooters, gran cantidad de juegos tripe A están diseñados con este motor.

2.2.3 Unity



Ilustración 4. Unity Engine

Probablemente el mejor o uno de los mejores engines de desarrollo para 2D y 3D. Es gratuito, las licencias de pago son asequibles, permite exportación a una gran cantidad de sistemas y ofrece soporte para Directx 11. Al igual que Unreal Engine tiene una curva de aprendizaje elevada debido a que es posible diseñar cualquier videojuego en el debido a su potencia. Sin duda la solución óptima junto a Unreal Engine para un proyecto de envergadura.

2.2.4 Construct3



Ilustración 5. Construct3 Engine

Este engine es un motor de desarrollo híbrido en el cual se puede crear diversos tipos de videojuegos, desde plataformas hasta PRG. Está orientado al desarrollo de juegos 2D y no requiere conocimientos profundos de programación, pero si tener la lógica. Es un motor perfecto para iniciarse.

2.3 Valoración económica del trabajo

La distribución de los costes tiene 3 espacios bien diferenciados, en ellos se comentarán los costes asociados a cada uno.

2.3.1 Software

Se ha utilizado una licencia de Windows de GameMaker comprada en Steam con un coste de **83 €**

Se ha utilizado el paquete Adobe, en concreto las herramientas Photoshop e Illustrator. Ambas licencias tienen un coste de 60 € al mes. Con un proceso de desarrollo de 3 meses son **180 €**

Coste de Software: 263 €

2.3.2 Hardware

El equipo utilizado para el desarrollo excede las características recomendadas para ambos programas. Sin embargo, el coste medio de un equipo completo de hardware que permita utilizar ambas plataformas es de **800 €**

Coste de Hardware: 800 €

2.3.3 Recursos Humanos

Tal y como se comentaba previamente estos proyectos son desarrollados por un equipo multidisciplinar, aunque no haya sido este el caso. Este equipo está

formado por una serie de integrantes con unos honorarios diferentes. Aproximadamente el coste de un proyecto de este tipo es el siguiente:

Campo	Coste / Hora	Número de Horas	Coste Total
Animador	35 €	100	3500 €
Diseñador de niveles	25 €	100	2500 €
Artista de diseño	25 €	100	2500 €
Programador	35 €	150	5250 €
Técnico de sonido	25 €	20	500 €
Especialista en marketing	20 €	20	400 €

Ilustración 6. Tabla de costes en recursos humanos

Coste de Recursos Humanos: 14.650 €

Con todos estos datos se puede concluir aproximadamente el coste de un proyecto como este y es de:

Coste de Software	263 €
Coste de Hardware	800 €
Coste de Recursos Humanos	14.650 €
COSTE TOTAL	15.713 €

2.4 Plataforma de destino

La idea inicial del proyecto iba a ser un lanzamiento dedicado únicamente para dispositivos móviles debido a la simpleza de sus mecánicas que se adaptaban a estos dispositivos. Sin embargo, tras un primer contacto con la plataforma de desarrollo se descartó esta opción debido a los escasos conocimientos de la misma y, a la poca experiencia en el este tipo de desarrollos.

Finalmente, la plataforma escogida será PC Windows por ser la más versátil y que se adaptaba de la mejor manera a los recursos de los que disponemos para este desarrollo.

3. Definición del juego

3.1 Historia y ambientación

El héroe es un guerrero que llega a la tierra de Damaris atraído por una voz de otro mundo que le dice que se tiene que convertir en el héroe de la misma. Una vez allí, le encomiendan encontrar al jefe de la Hermandad. Éste, harto de entrenar a cientos de guerreros sin valía, sigue buscando al merecedor de esta tierra. A medida que el héroe demuestra sus habilidades y completa las misiones será entrenado hasta la pelea final con el malvado Deimos.

Durante esta aventura el héroe deberá conocer el porqué de su llegada al reino, así como la identidad de la voz misteriosa que lo despierta. Esta no es más que su padre, que se encuentra cuidando de él mientras supera una enfermedad.

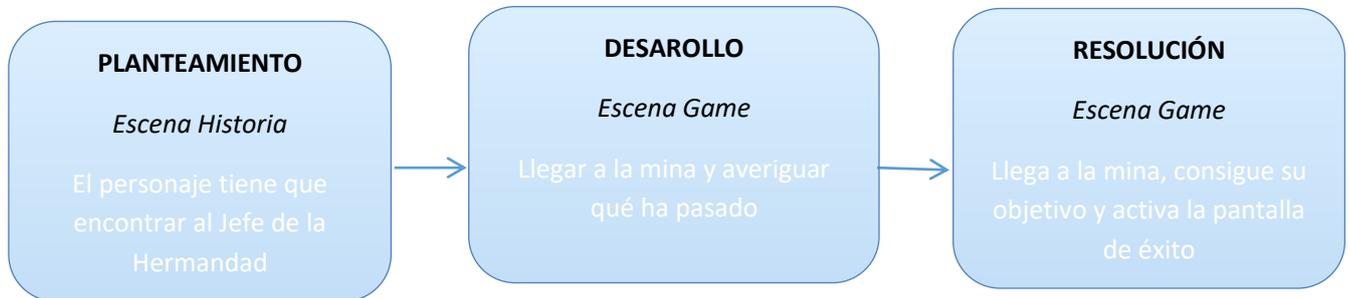
La historia de nuestro héroe, es la de cualquier chico que tiene un héroe dentro y puede junto con el apoyo de sus familiares, superar cualquier dificultad, porque todos son nuestros pequeños héroes.

3.1.1 Desarrollo del primer nivel

La trama inicial es sencilla, no tiene una narrativa compleja, lo que tendrá es que esquivar la horda de esqueletos que vagan por el campo hasta llegar a la ciudad. Una vez allí conocerá al jefe de la hermandad que le instará a averiguar lo ocurrido en la mina y recuperar la gema que se encuentra dentro de ella. Finalmente, después de despejar la mina y confrontar al jefe de espadas, recuperará la gema y acabará el primer nivel.

- **Planteamiento:** Insta al héroe a llegar a la ciudad y encontrar al jefe de la hermandad

- **Desarrollo:** El jefe le pide despejar el misterio sobre la mina y le entrega la llave de la misma.
- **Resolución:** Consigue acabar con el maestro de espadas quien posee la gema y acaba el nivel



3.2 Objetivos planteados al jugador

Antes del comienzo del juego, el jugador encontrará un mensaje con un objetivo. Un mensaje de aliento y fuerza. Esta voz acompañará al personaje durante la

Despierta... Despierta... DESPIERTA!!!

*La gente de esta tierra está esperando a alguien como tú.
Confía en tus habilidades, tú eres muy capaz.*

Te estaré vigilando y, llegado el momento, acudiré en tu ayuda.

*Siempre fuiste **MI PRIMER HÉROE***

historia y ayudará a la narrativa para explicar el motivo de por qué el héroe se encuentra en la tierra de Damaris. En concreto, al principio y final de cada nivel.

Los objetivos planteados de forma recurrente en cada nivel será despejar una zona de enemigos y confrontar al jefe de cada mazmorra. Conforme la aventura avanza los enemigos que nos encontraremos contarán con mejores habilidades,

así como el propio héroe que, gracias a los entrenamientos del jefe de la hermandad obtendrá nuevas habilidades con la espada e incluso mágicas.

3.3 Personajes

El héroe es el personaje principal y será guiado en todo momento por el jugador. El protagonista de la historia podrá caminar, correr, pelear con su espada y estar parado cuando lo crea necesario.

El avance en la historia tiene lugar gracias a las interacciones con otros personajes de la historia. Estas conversaciones le darán las instrucciones necesarias para conseguir el objetivo del juego.

3.3.1 Personajes y elementos de interacción con el Héroe

- **La voz:** El héroe es despertado por una voz que le dice que él es el elegido para salvar a la tierra de Damaris.
- **El viejo:** Personaje que lo recibe al inicio del juego y le indica que su primer objetivo es hablar con el jefe de la Hermandad quien le espera ansioso.
- **Jefe de la Hermandad:** Entrenador de muchos guerreros y ahora de este héroe que tiene que demostrar sus habilidades. El jefe pone todas sus esperanzas en él y le pide diversas tareas para que finalmente derrote al malvado Deimos.
- **Enemigos:** puede acabar con ellos para conseguir monedas o corazones de vida.
- **Baúles:** Puede abrirlos para buscar las llaves de las mazmorras
- **Carteles:** Muestran al héroe información relativa a las zonas.
- **Monedas y corazones:** Recuperan salud y aumentan la cantidad de monedas totales del héroe que en capítulos posteriores podrá usar para comprar objetos en la tienda
- **Animales:** Proporcionan corazones y monedas
- **Plantas:** Proporcionan corazones y monedas

Todo este tipo de interacciones serán explicadas de forma más detallada en los capítulos de diseño técnico y diseño de niveles debido a la gran relación que tienen a nivel lógico y de diseño.

3.4 Concept Art

Mostramos a continuación artes conceptuales de los principales actores del desarrollo actual del producto.

3.4.1 El Héroe



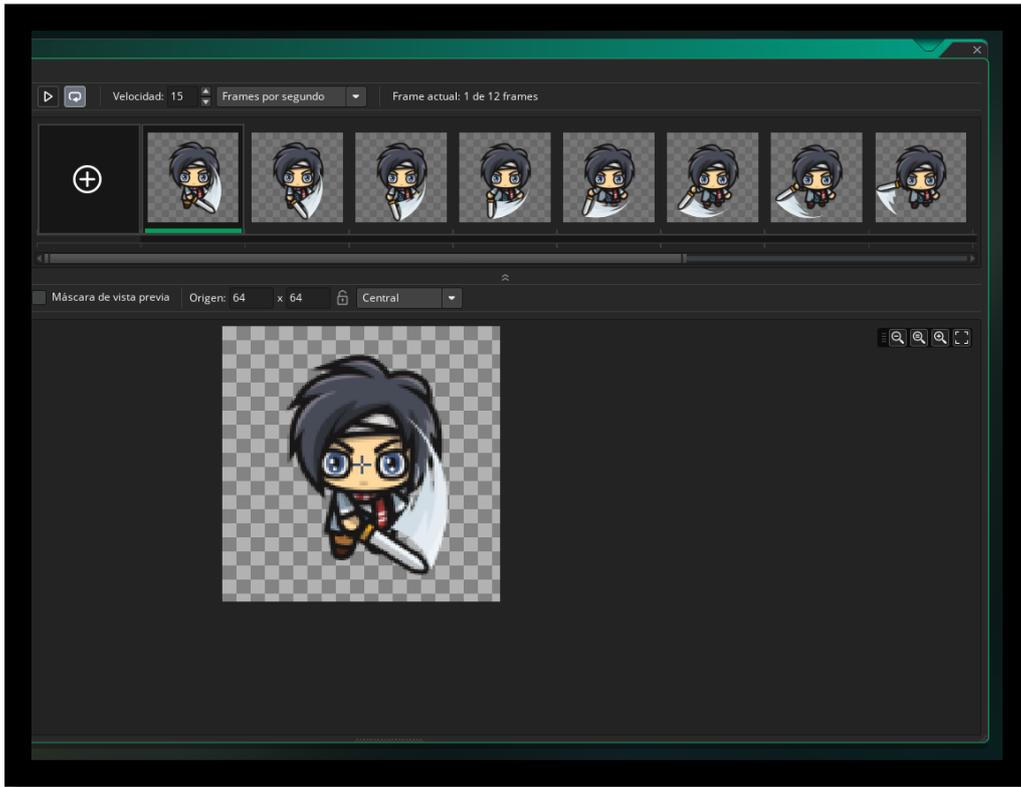


Ilustración 8. Arte del protagonista

3.4.2 El jefe de la hermandad

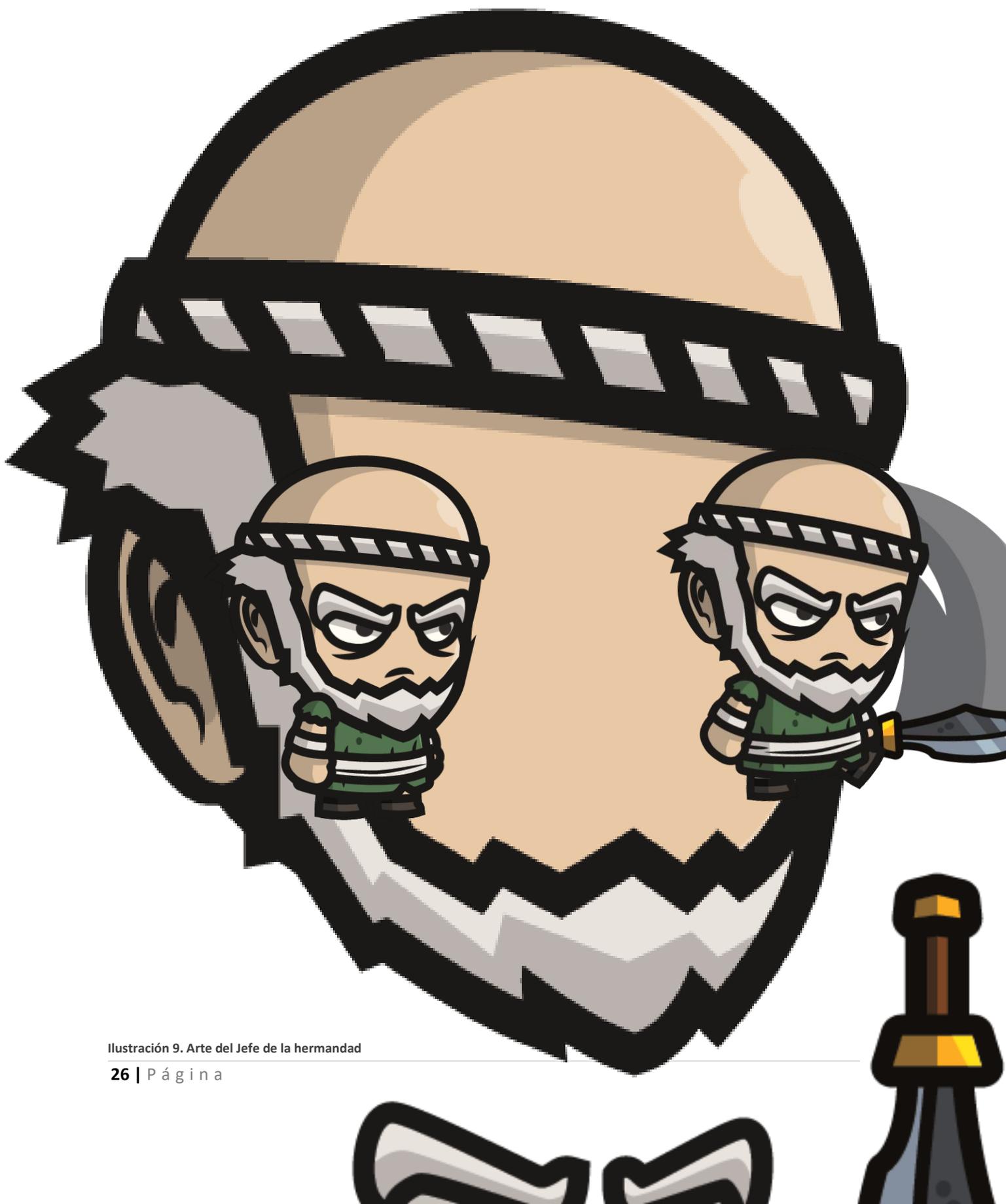


Ilustración 9. Arte del Jefe de la hermandad

3.4.3 Enemigos

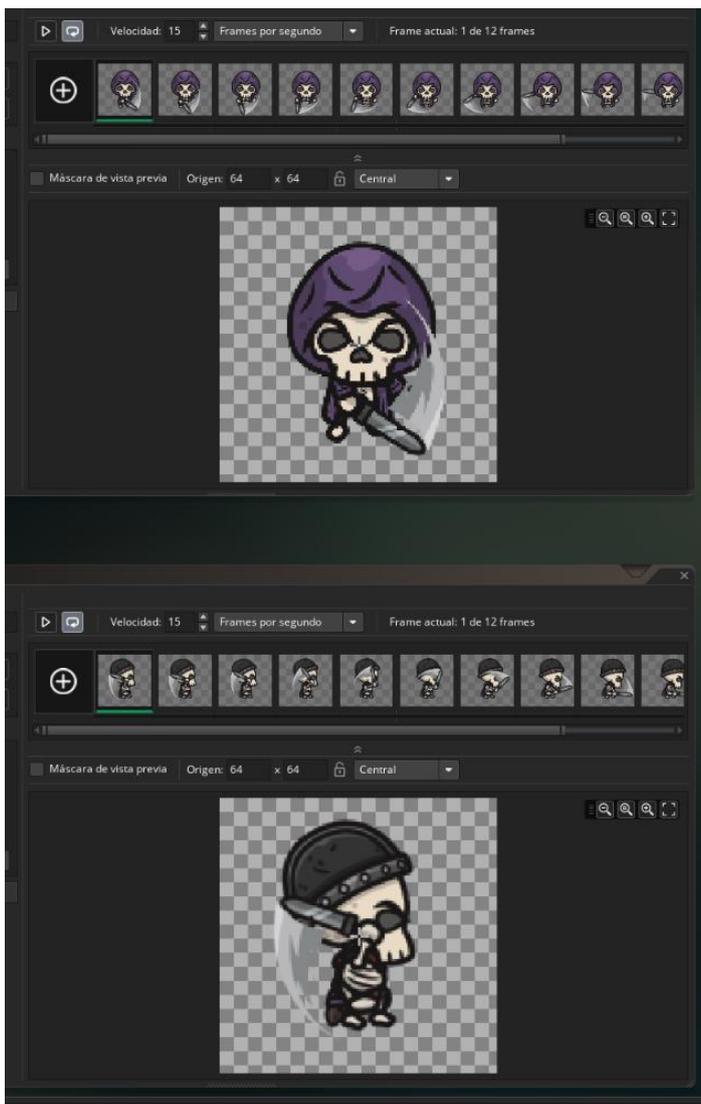


Ilustración 10. Arte de enemigos

4. Diseño técnico

4.1 Entorno escogido y requisitos



Ilustración 11. Plataforma escogida

El entorno escogido para este proyecto entre los propuestos inicialmente ha sido **GameMaker**. A continuación, se comentarán los aspectos clave que llevaron a la toma de esta decisión:

- **Fácil e intuitivo para principiantes:** El primer gran hándicap del proyecto era este. No existían conocimientos previos en este tipo de desarrollos por lo que era necesario aportar todas las facilidades posibles y una de las mayores ventajas de GameMaker es esta.
- **Ideal para desarrollo de juegos en 2D:** Es cierto que para juegos 3D GameMaker sufre, pero gracias al editor interno de sprites y al diseño de rooms, se postulaba como una opción adecuada para el tipo de proyecto inicial.

- **Acceso a la versión de pago asequible:** Se pudo adquirir la versión de pago de la plataforma de desarrollo en Windows a un precio irrisorio.

Se realizó un estudio de todas las plataformas y aunque Unity fue la candidata inicial, finalmente, se escogió esta.

Esta versión de GameMaker está disponible para Windows y Mac y en concreto hemos usado “GameMaker Studio 2 Desktop”.



Ilustración 12. Requisitos plataforma

Los requisitos son los siguientes:

MÍNIMO:

Requiere un procesador y un sistema operativo de 64 bits

SO: Microsoft 64bit Windows 7

Procesador: 64bit Intel compatible Dual Core CPU

Memoria: 2 GB de RAM

Gráficos: DX11 compliant graphics card

DirectX: Versión 11

Red: Conexión de banda ancha a Internet

Almacenamiento: 3 GB de espacio disponible

RECOMENDADO:

Requiere un procesador y un sistema operativo de 64 bits

SO: Microsoft 64bit Windows 10

Procesador: 64bit Intel compatible Quad Core CPU

Memoria: 8 GB de RAM

Gráficos: DX11 based graphics card

DirectX: Versión 11

Red: Conexión de banda ancha a Internet

Almacenamiento: 3 GB de espacio disponible

4.2 Herramientas utilizadas

Además de la versión de GameMaker comentada anteriormente se han utilizado otras herramientas para la edición:

4.2.1 Adobe Photoshop CS6



Ilustración 13. Adobe Photosthop Logo

Photoshop es un editor de mapas de bits, y uno de los programas más famosos utilizado para cualquier tipo de diseño, en especial los gráficos.

Se ha utilizado Photoshop para retoques en los sprites y, para el diseño de manuales y animaciones. Aunque el editor de GameMaker permitía algunas modificaciones no era muy potente y se ha requerido el uso de Photoshop en multitud de ocasiones.



Ilustración 14. Diseño en Photoshop

4.2.2 Adobe Premiere Pro



Ilustración 15. Adobe Premiere Pro

Premiere Pro es un software de edición de video que permite el montaje y edición de los mismos. Es el más popular del mercado y consigue unos resultados realmente profesionales.

Se ha usado este software para la edición del video de entrega y presentación del proyecto.

4.3 Inventario de Assets y Recursos del juego

Todos los recursos usados en “My First Hero” son de licencia privada y fueron adquiridos en el paquete Humble Bundle llamado “RPG Game Dev” en noviembre de 2018. La decisión de adquirir dicho paquete fue la escasa cantidad de recursos disponibles y su elevado coste por separado. Esto, unido al hecho de que este paquete se encontraba a un precio magnífico fueron los que propiciaron su adquisición.



Ilustración 16. Humble Bundle RPG Pack

No todos los contenidos del paquete fueron usados debido a que comprendían temáticas muy distintas. El uso que se ha dado de dicho paquete ha sido para 2 fuentes de recursos: Los sprites y los sonidos. A continuación, realizaremos un listado de los elementos usados de este paquete en nuestro proyecto.

4.3.1 Sprites

Se han utilizado más de 200 sprites diferentes para este proyecto, que incluyen las animaciones de los personajes y del entorno, así como el diseño de menús y elementos de la interfaz.

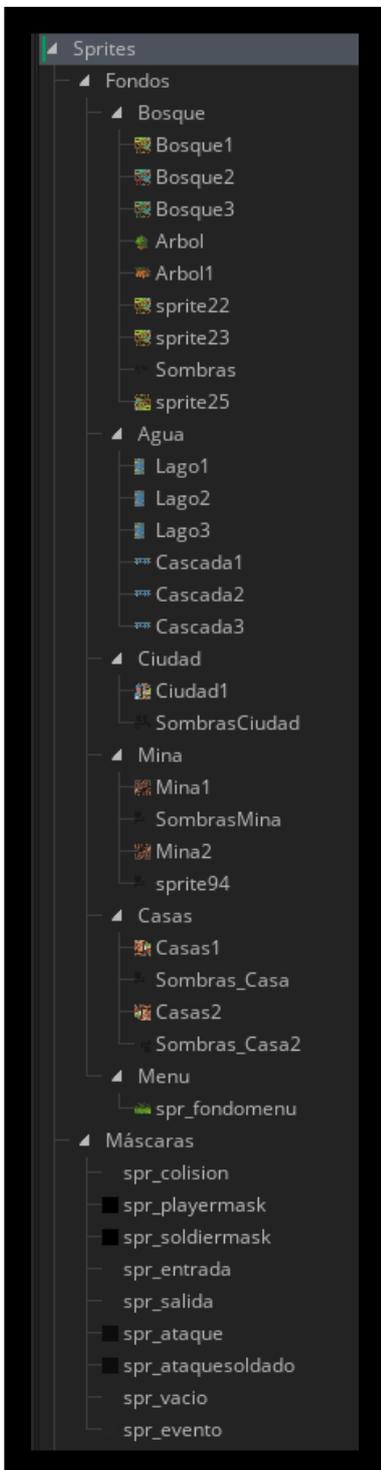


Ilustración 18. Lista de sprites

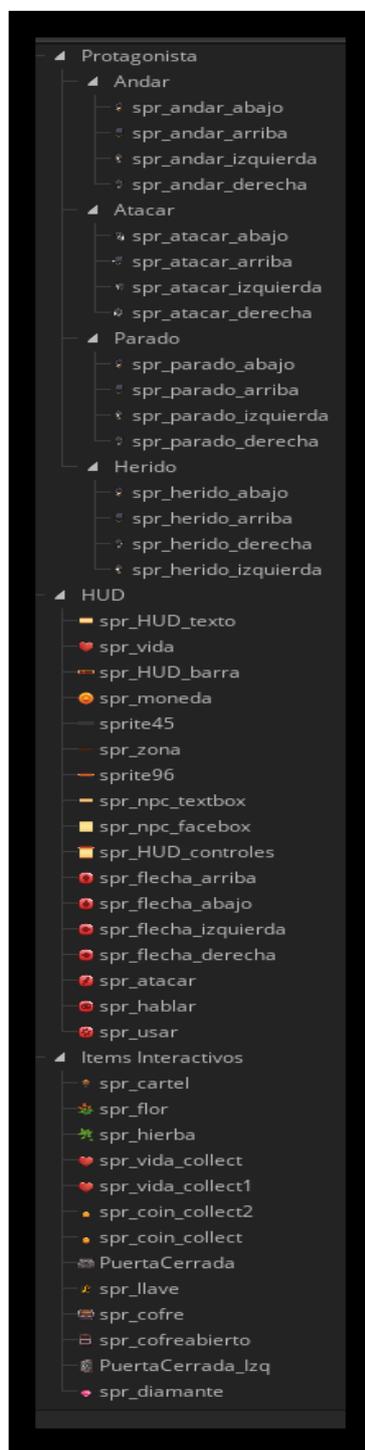


Ilustración 17. Lista de sprites



Ilustración 19. Lista de sprites

En las listas previas se encuentran aproximadamente la mitad de los sprites utilizados. Se han utilizado todos los incluidos en el proyecto.

4.3.2 Sonido

Se han utilizado tanto efectos de sonido como piezas musicales para los fondos. Los recursos utilizados son los siguientes:



Ilustración 20. Lista de sonidos incluidos

El inventario completo del juego dada su longitud puede observarse en el proyecto del mismo.

4.4 Arquitectura del juego | Diseño

En este capítulo se comentarán detalladamente los procesos de diseño de la arquitectura del juego, desde las propias animaciones de los personajes, a su implementación. Como se detallará posteriormente, el diseño de mapas es crucial en este proyecto por lo que tiene reservado un capítulo propio y, en él, se detallarán exclusivamente las decisiones y tipos de recursos usados en la propuesta.

Dado que el objetivo de esta memoria no es la de ofrecer un código comentado del proyecto, sino ofrecer el punto de vista del desarrollador en las decisiones tomadas, solo se incluirán pequeñas piezas que por motivos esenciales es necesario comentar.

4.4.1 Animación y diseño de personajes

Existen 30 personajes distintos en el primer nivel de My First Hero, todos con sus propias animaciones. Para el diseño de las animaciones se han usado los sprites de la colección y se han colocado para dar forma a la misma.



Ilustración 22. Animación del héroe

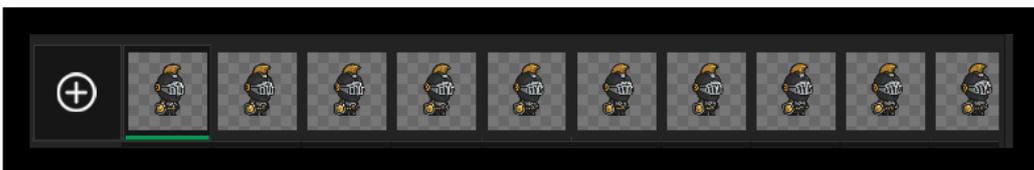


Ilustración 21. Animación del guardia real

El proceso de animación como podemos ver consisten en unir en un mismo sprite la secuencia de imágenes que conforman el movimiento, una vez unidos se realiza la nomenclaturización y se añade al proyecto para su posterior uso.

Además de las animaciones propias de los personajes se ha llevado a cabo un trabajo paralelo para el sistema de conversación. Se han incluido líneas de diálogo para todos los personajes en pantalla y se diseñaron sprites personalizados para su retrato en el HUD.

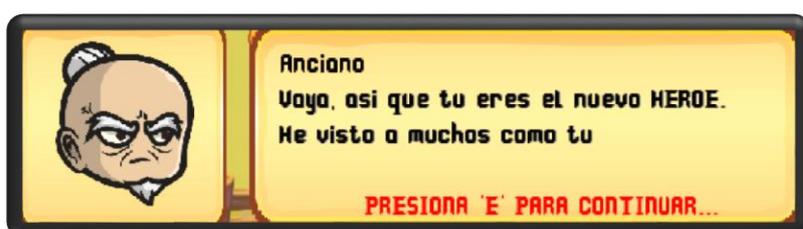


Ilustración 23. Conversación anciano HUD



Ilustración 24. Conversación chico HUD

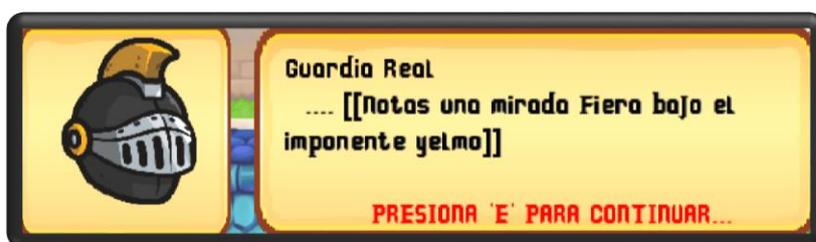


Ilustración 25. Conversación guardia real HUD

Las animaciones diseñadas en el proyecto varían en función de su impacto en el mismo. Para los NPC's no era necesario realizar animaciones tan detalladas como para el protagonista o los enemigos debido a que no realizarían las mismas acciones. Sin embargo, se estimó oportuno dotar a todos los personajes de al menos una animación de estado parado para generar una sensación de entorno vivo.

4.4.2 Lógica del personaje

La parte principal de la aventura también es la que conlleva un diseño lógico mayor. Para entender el funcionamiento de enemigos se ha seguido una lógica aplicada al personaje que se hace extensible a los mismos. Para poder realizar cambios entre las diversas posibilidades del héroe se ha diseñado una máquina de estados que funciona de la siguiente manera:

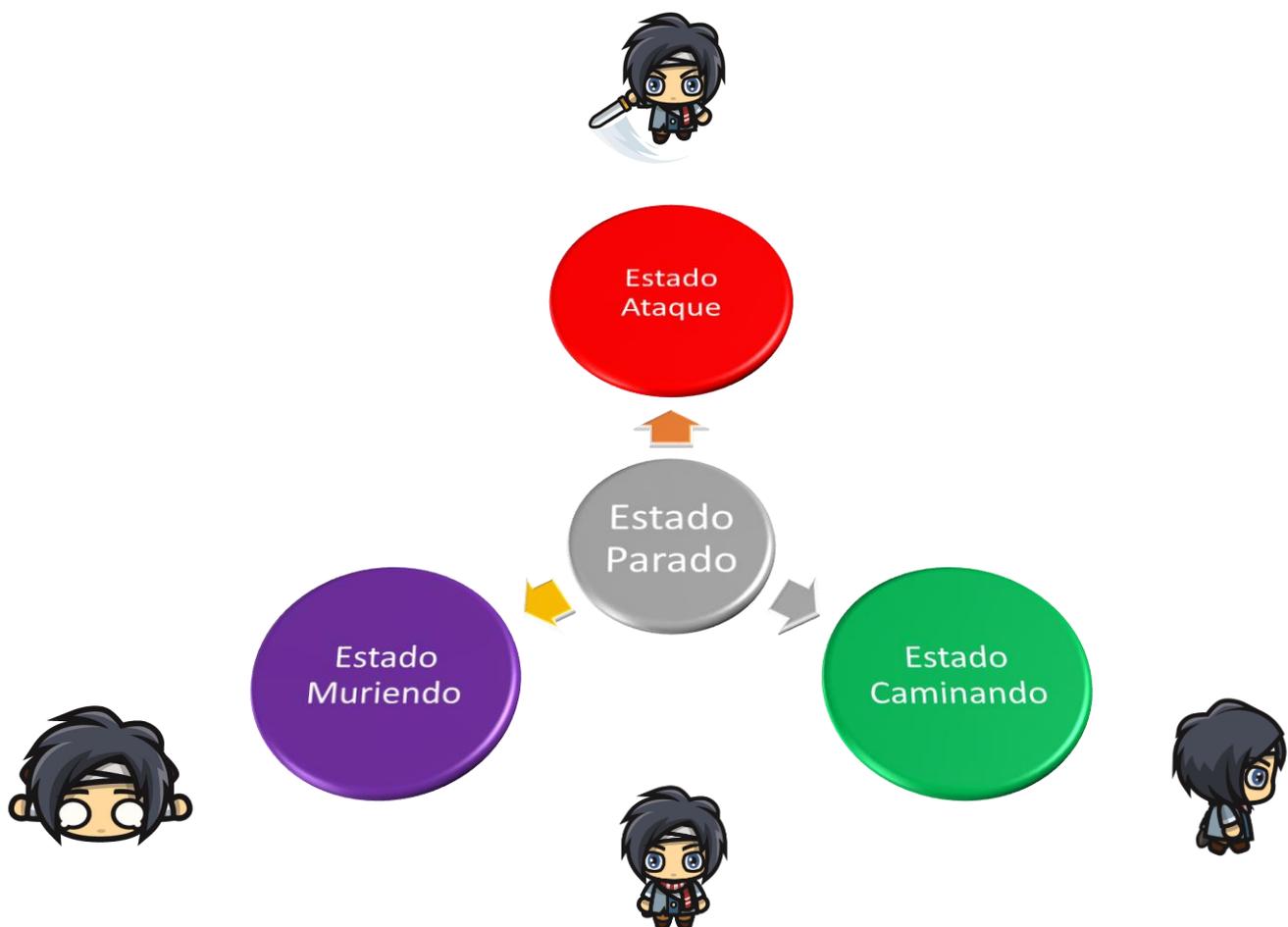


Ilustración 26. Estados Héroe

En función de la dirección y de la acción que realicemos obtenida a través del teclado como input realizaremos el cambio de estado. Estos cambios de estado se ejecutan mediante un script personalizado.

Este es el código comentado del objeto protagonista y sus respectivos scripts que conforman el diseño completo del mismo. Es necesario puntualizar que nuestro objeto héroe funciona en consonancia con un objeto llamado controlador que actual como director de película, y controla todo lo que ocurre en el entorno.

4.4.2.1 Código del héroe

4.4.2.1.1 Create Event

```
image_speed = 0;
```

```
// Estados del personaje
```

```
enum state {
```

```
    parado = player_idle,
```

```
    caminando = player_walking,
```

```
    muriendo = player_dying,
```

```
    atacando = player_attacking
```

```
}
```

```
dir = "abajo"; // Dirección por defecto del personaje
```

```
estado = state.parado; // Estado por defecto
```

```
salida = -1; // Por defecto no salimos hacia ninguna room
```

```
crear_ataque = true; // Para crear el objeto ataque que permite realizarlo
```

```
global.debug = false;
```

```
vida = 2;
```

```
vida_maxima = 3;
```

```
monedas = 0;
```

```
puntos = 0;
```

```
llaves = 0;
```

```
llavesclave [0] = false;
```

```
llavesclave [1] = false;
```

```
llavesclave [2] = false;
```

4.4.2.1.2 Step Event

```
// Inicialización de velocidades
```

```
vv= 0;
```

```
vh= 0;
```

```
v = 4;
```

```
vr = 6;
```

```
depth = -y;
```

// Ajuste de Velocidades

```
if (keyboard_check(vk_shift)){
```

```
    image_speed = 1.60;
```

```
    vh= (-keyboard_check(vk_left) + keyboard_check(vk_right)) * vr;
```

```
    vv= (-keyboard_check(vk_up) + keyboard_check(vk_down)) * vr;
```

```
}else{
```

```
    image_speed = 0.80;
```

```
    vh= (-keyboard_check(vk_left) + keyboard_check(vk_right)) * v;
```

```
    vv= (-keyboard_check(vk_up) + keyboard_check(vk_down)) * v;
```

```
}
```

// GESTIÓN DEL MOVIMIENTO

//Determinar dirección

```
if (vv > 0) dir = "abajo";
```

```
else if (vv < 0 ) dir = "arriba";
```

```
else if (vh > 0 ) dir = "derecha";
```

```
else if (vh < 0 ) dir = "izquierda";
```

```
// Si no estamos atacando comprobamos si estamos parados o caminando
```

```
if (estado != state.atacando){
```

```
// Si existe dirección establecemos el estado caminar
```

```
if (dir != "") estado = state.caminando;
```

```
// Si no nos movemos nos quedamos parados
```

```
if (vh == 0 and vv==0) estado = state.parado;
```

```
} else {
```

```
    vh = 0;
```

```
    vv = 0;
```

```
}
```

```
// GESTION DE ATAQUE
```

```
// Si pulsamos atacar y no estamos atacando actualmente
```

```
if (keyboard_check_pressed(vk_space) and estado != state.atacando){
```

```
// Cambiamos el estado y reproducimos los sonidos de ataque
```

```
estado = state.atacando;
```

```
audio_play_sound(tajo1,10,false);
```

```

        audio_play_sound(choose(grito1,grito2,grito3,grito4),5,false);

        image_index = 0;

    }

    // GESTION DE MUERTE

    if (vida <= 0) {

        estado = state.muriendo;

        vh = 0;

        vv = 0;

        global.new_music = msc_gameover;

    }

    // GESTIÓN DE COLISIONES

    if (instance_place(x+vh,y,obj_colision)){

        vh=0;

    }

    if (instance_place(x,y+vv,obj_colision)){

        vv=0;

    }

    // Colisión horizontal con objetos interactivos

```

```
interactivo = instance_place(x+vh,y,obj_interactivo);  
if(interactivo) {  
    if(interactivo.colision == true){  
        vh = 0;  
    }  
}
```

// Colision vertical con objetos interactivos

```
interactivo = instance_place(x,y+vv,obj_interactivo);  
if(interactivo) {  
    if(interactivo.colision == true){  
        vv = 0;  
    }  
}
```

// Colisión horizontal con objetos evento

```
evento = instance_place(x+vh,y,obj_event);  
if(evento) {  
    if(evento.colision == true){  
        vh = 0;  
    }  
}
```

```

        global.contacto = true;
    }
}

// Colision vertical con objetos evento

evento = instance_place(x,y+vv,obj_event);
if(evento) {
    if(evento.colision == true){
        vv = 0;
        global.contacto = true;
    }
}

// Colision con corazones,monedas y diamante

corazon = instance_place(x+vh,y+vv,obj_corazon);
if (corazon) {
    audio_play_sound(snd_corazon,10,false);
    vida = clamp(vida+1,0,vida_maxima);
    with (corazon) instance_destroy();
}

moneda = instance_place(x+vh,y+vv,obj_moneda);
if (moneda) {
    monedas ++;
}

```

```

    puntos = puntos + 10;

    audio_play_sound(snd_moneda,10,false);

    with (moneda) instance_destroy();
}

diamante = instance_place(x+vh,y+vv,obj_diamante);

if (diamante) {

    global.final = 0;

    puntos = puntos + 10000;

    audio_play_sound(diamanteconseguido,10,false);

    with (diamante) instance_destroy();

}

```

// Cambio de room

```

entrada = instance_place(x,y,obj_entrada);

with (entrada) {

    if (room_exists(mapa)){

        other.salida = salida;

        room_goto(mapa);

    }else{

        show_message("No existe la room: " + entrada.mapa);

    }

}

x += vh;

```

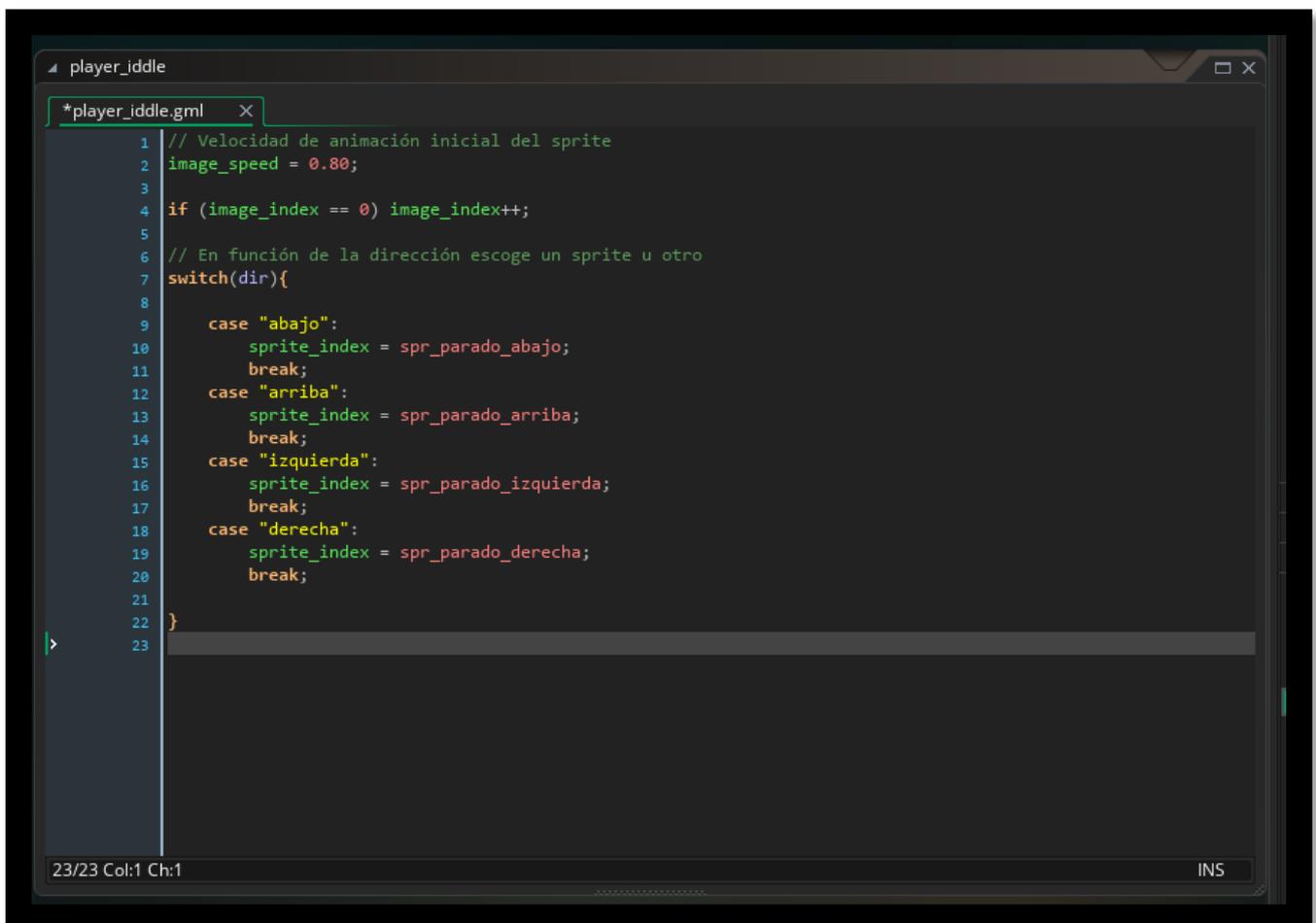
y += vv;

script_execute(estado); // *Ejecutamos el script del estado actual*

Con esta lógica tenemos implementado al personaje. Ahora comentaremos los scripts que ejecuta en cada estado.

4.4.2.1 Código de scripts del héroe

4.4.2.1.1 Estado Parado

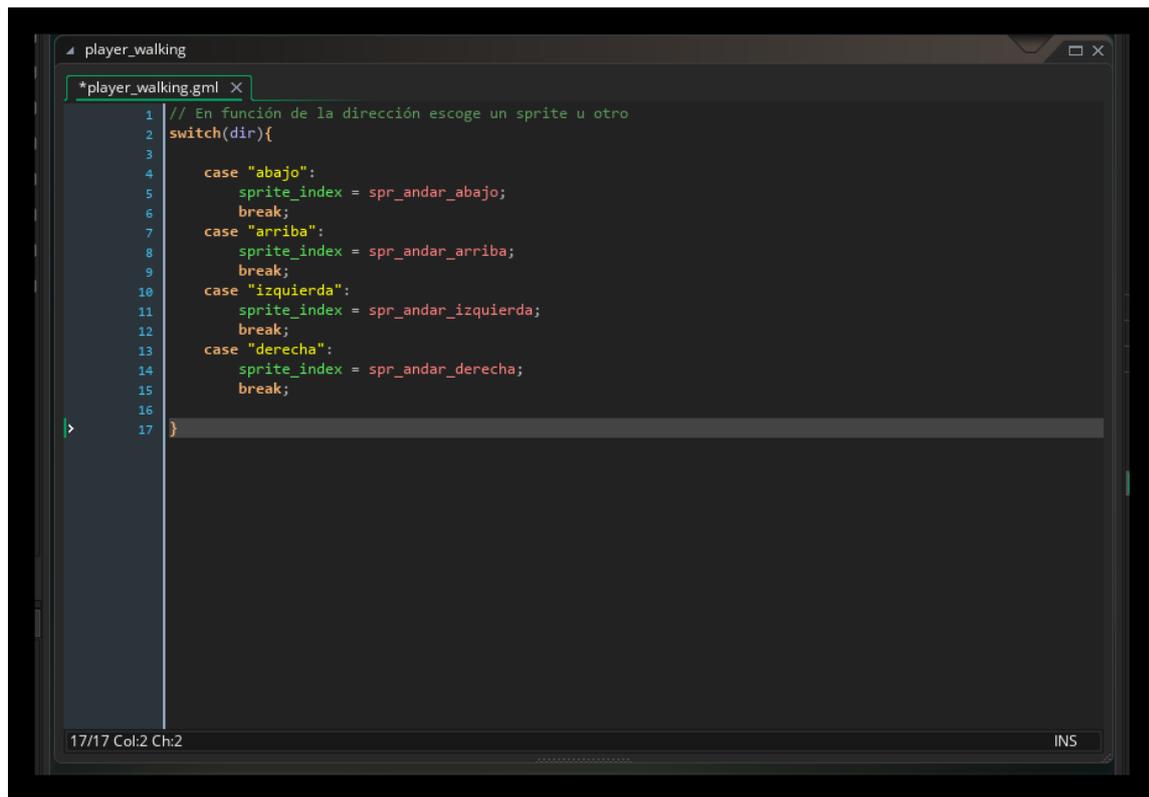
A screenshot of a code editor window titled 'player_idle'. The editor shows a GML script file named '*player_idle.gml'. The code is as follows:

```
1 // Velocidad de animación inicial del sprite
2 image_speed = 0.80;
3
4 if (image_index == 0) image_index++;
5
6 // En función de la dirección escoge un sprite u otro
7 switch(dir){
8
9     case "abajo":
10        sprite_index = spr_parado_abajo;
11        break;
12
13    case "arriba":
14        sprite_index = spr_parado_arriba;
15        break;
16
17    case "izquierda":
18        sprite_index = spr_parado_izquierda;
19        break;
20
21    case "derecha":
22        sprite_index = spr_parado_derecha;
23        break;
24
25 }
```

The editor interface includes a tab for the file, a line number column on the left, and a status bar at the bottom showing '23/23 Col:1 Ch:1' and 'INS'.

Ilustración 27. Sprite parado héroe

4.2.1.2 Estado Caminando

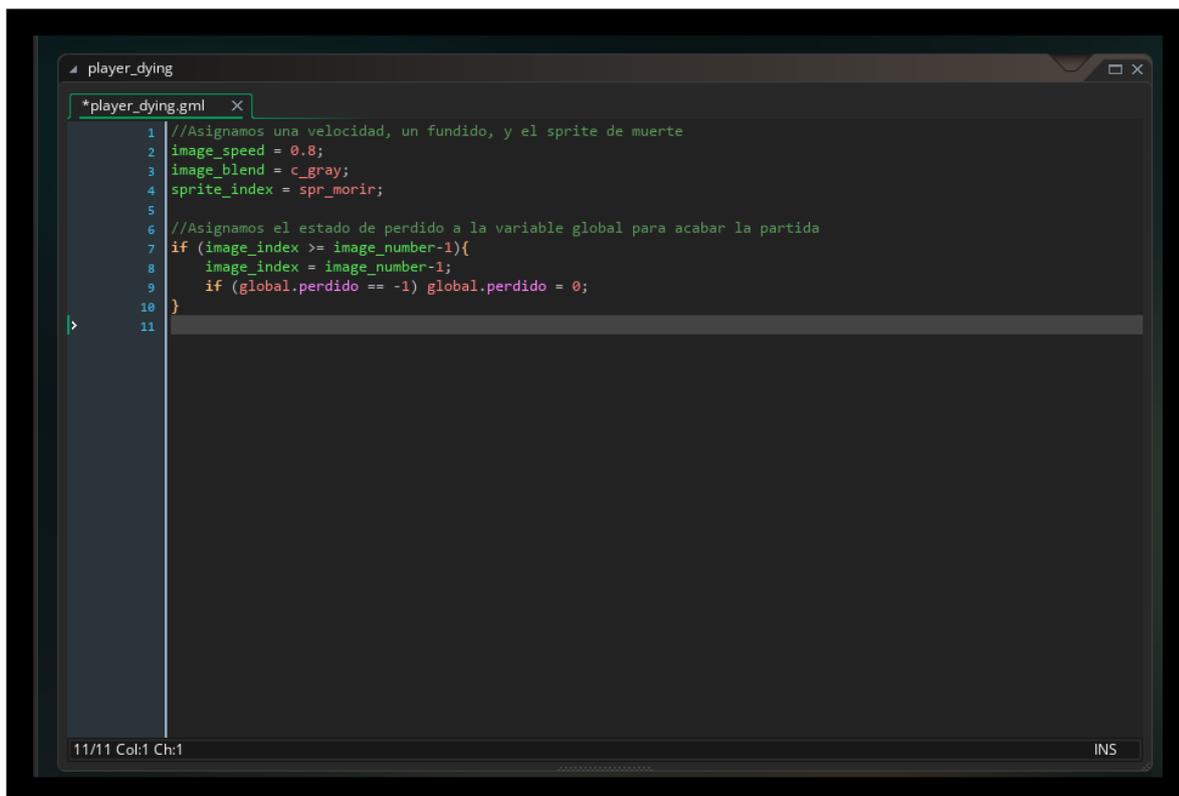


```
player_walking
*player_walking.gml X
1 // En función de la dirección escoge un sprite u otro
2 switch(dir){
3
4     case "abajo":
5         sprite_index = spr_andar_abajo;
6         break;
7     case "arriba":
8         sprite_index = spr_andar_arriba;
9         break;
10    case "izquierda":
11        sprite_index = spr_andar_izquierda;
12        break;
13    case "derecha":
14        sprite_index = spr_andar_derecha;
15        break;
16
17 }
```

17/17 Col:2 Ch:2 INS

Ilustración 28. Sprite andando héroe

4.4.2.1.3 Estado Muriendo

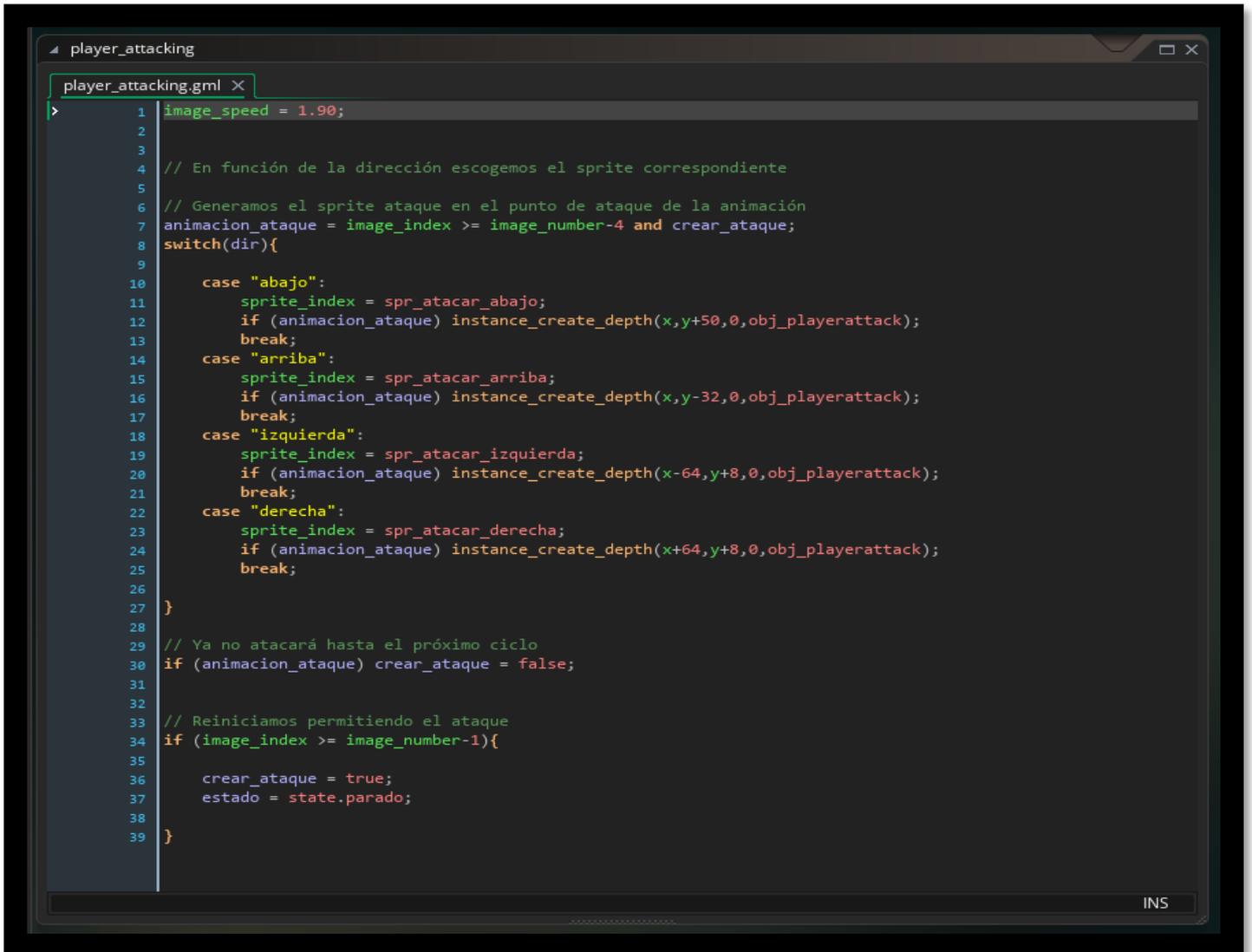


```
player_dying
*player_dying.gml X
1 //Asignamos una velocidad, un fundido, y el sprite de muerte
2 image_speed = 0.8;
3 image_blend = c_gray;
4 sprite_index = spr_morir;
5
6 //Asignamos el estado de perdido a la variable global para acabar la partida
7 if (image_index >= image_number-1){
8     image_index = image_number-1;
9     if (global.perdido == -1) global.perdido = 0;
10 }
11
```

11/11 Col:1 Ch:1 INS

Ilustración 29. Sprite muriendo héroe

4.4.2.1.4 Estado Atacando

A screenshot of a code editor window titled 'player_attacking'. The code is written in GML and defines the logic for a player's attack state. It starts with 'image_speed = 1.90;'. A comment indicates that the sprite is chosen based on direction. The code then sets 'animacion_ataque' to true if the animation index is within the range of the attack animation. A switch statement handles four directions: 'abajo' (down), 'arriba' (up), 'izquierda' (left), and 'derecha' (right). For each direction, it sets a specific 'sprite_index' and creates an 'obj_playerattack' instance at a defined depth. After the switch, it sets 'crear_ataque = false;'. A final if statement resets 'crear_ataque = true;' and 'estado = state.porado;' when the animation index reaches the end of the animation sequence.

```
1 image_speed = 1.90;
2
3
4 // En función de la dirección escogemos el sprite correspondiente
5
6 // Generamos el sprite ataque en el punto de ataque de la animación
7 animacion_ataque = image_index >= image_number-4 and crear_ataque;
8 switch(dir){
9
10  case "abajo":
11     sprite_index = spr_atacar_abajo;
12     if (animacion_ataque) instance_create_depth(x,y+50,0,obj_playerattack);
13     break;
14  case "arriba":
15     sprite_index = spr_atacar_arriba;
16     if (animacion_ataque) instance_create_depth(x,y-32,0,obj_playerattack);
17     break;
18  case "izquierda":
19     sprite_index = spr_atacar_izquierda;
20     if (animacion_ataque) instance_create_depth(x-64,y+8,0,obj_playerattack);
21     break;
22  case "derecha":
23     sprite_index = spr_atacar_derecha;
24     if (animacion_ataque) instance_create_depth(x+64,y+8,0,obj_playerattack);
25     break;
26
27 }
28
29 // Ya no atacará hasta el próximo ciclo
30 if (animacion_ataque) crear_ataque = false;
31
32
33 // Reiniciamos permitiendo el ataque
34 if (image_index >= image_number-1){
35
36     crear_ataque = true;
37     estado = state.porado;
38
39 }
```

Ilustración 30. Sprite atacando héroe

Es necesario comentar en este apartado que en el momento del ataque se crea una instancia del objeto playerattack. Este objeto será el que realice las interacciones con los otros objetos del juego y, en función de con cuál de ellos impacte generará un resultado u otro.

Estos scripts se utilizan de la misma manera para los enemigos y tienen el mismo diseño con la única diferencia de los sprites asignados, como es lógico.

Además de estos enemigos existen objetos interactivos en el escenario de los cuales hablaremos posteriormente en el diseño de mapas, pero resulta interesante definir su funcionamiento.

Como punto final hablando del personaje, enemigos y en general de todos los objetos interactivos y destructibles. Todos ellos comparten un atributo que es la vida y que marca el devenir de estos objetos. Si alguno de ellos pierde todos sus puntos de vida o bien desaparece del juego o bien el juego se acaba si es el caso del jugador.

4.4.3 Objetos interactivos y colisiones

Debido a que es necesario delimitar los márgenes de nuestro mapeado se ha diseñado un evento colisión. Como se puede apreciar en el código del personaje, en el momento de contacto entre este objeto y el jugador las velocidades referentes al objeto del personaje quedan anuladas y no podrá atravesarlo.

Siguiendo esta misma lógica se diseñan los objetos interactivos. Estos cuentan con la funcionalidad de las colisiones, pero añaden una ejecución adicional si además de existir contacto el jugador pulsa la tecla asignada para la interacción, en este caso la tecla 'E' del teclado.

En caso de realizar dicha acción, se abrirán las líneas de texto correspondientes a dicho objeto. Esto es aplicable a **carteles, puertas, cofres y NPC's**.



Ilustración 32. Objeto interactivo cartel



Ilustración 31. Objeto interactivo puerta

4.4.4 Enemigos e IA enemiga

Los enemigos comparten muchas de las funcionalidades del héroe. Su sistema de estados es el mismo, así como la generación del objeto de ataque para atacar al jugador. La única diferencia como es obvio es que deben ser controlados por el PC. Vamos a plantear sin incluir código esta vez, como se ha realizado este diseño.

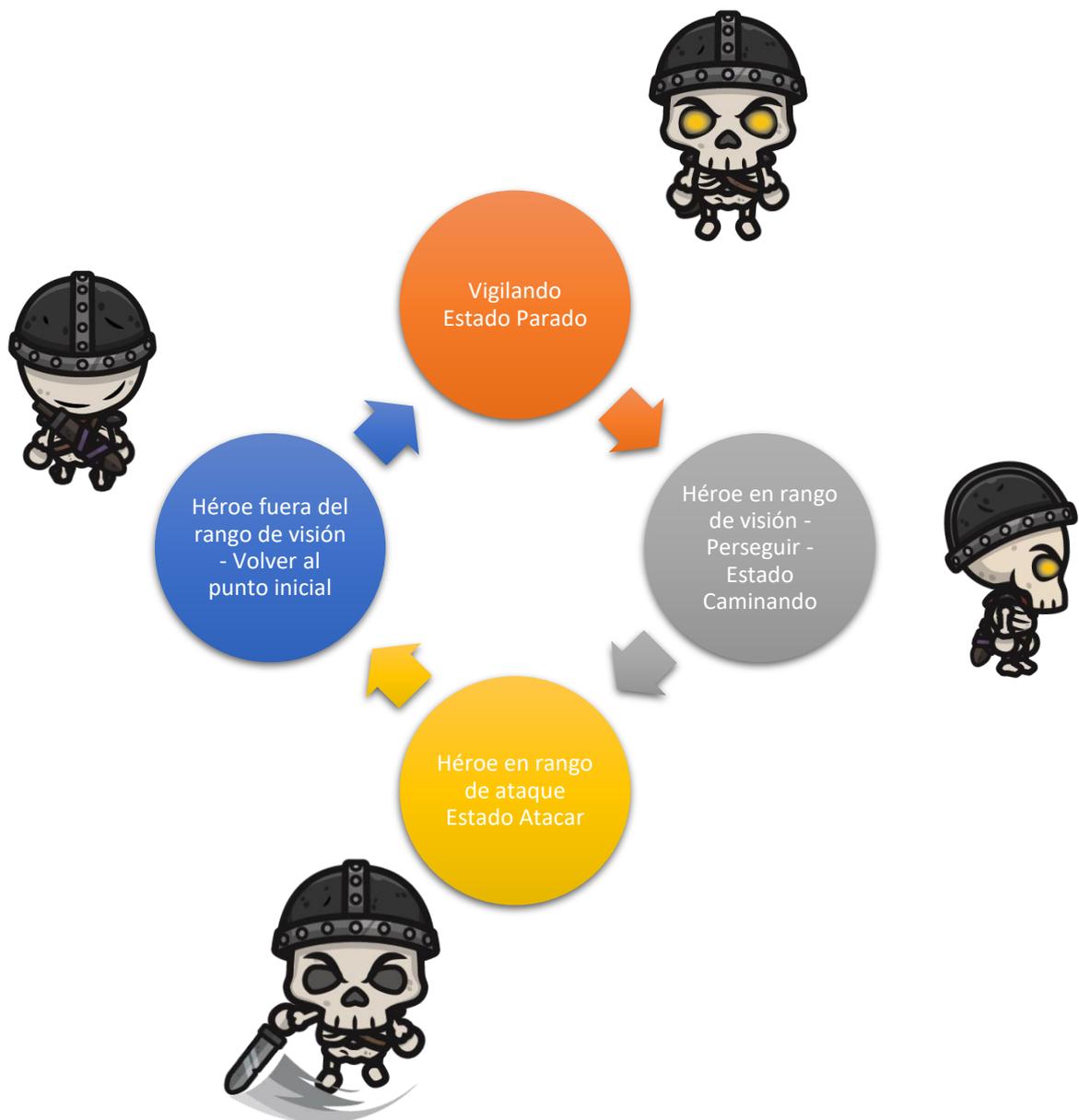


Ilustración 33. Estados posibles de los enemigos

Para realizar esta persecución establecemos un radio de visión, y realizamos una comprobación actualizada de si el objeto héroe está a una distancia inferior o igual a ese radio, en caso de ser así iniciamos la persecución y ejecutamos los scripts necesarios de la misma manera que el jugador.

La posición inicial es guardada antes de iniciar el movimiento por lo que si el jugador sale de este rango el enemigo volverá a dicha posición.

Sería interesante implementar un sistema de rutas para evitar que los enemigos fueran estáticos, pero en este proyecto fue imposible combinar ambas funcionalidades.

4.4.5 Objetos destructibles y animales

Los objetos destructibles como plantas son un híbrido entre los objetos NPC y los enemigos. Comparten las características de vida y colisión, pero estos, además, tienen interacción con el playerattack por lo que pueden perder vida. Al perder sus puntos de vida cambiarán el sprite y aparecerán cortadas, además de eliminar la colisión y permitir al jugador pasar por ese punto.

Los animales por su parte tienen una generación interna de posición aleatoria por lo que vagarán por el mapa y, al contacto con la espada del jugador, perderán sus puntos de vida y desaparecerán.

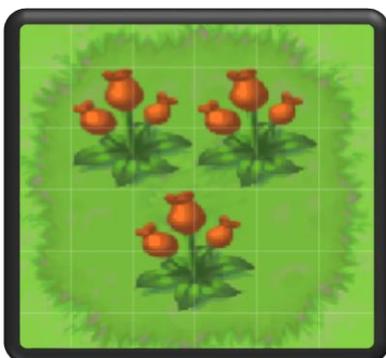


Ilustración 35. Objeto destructible planta

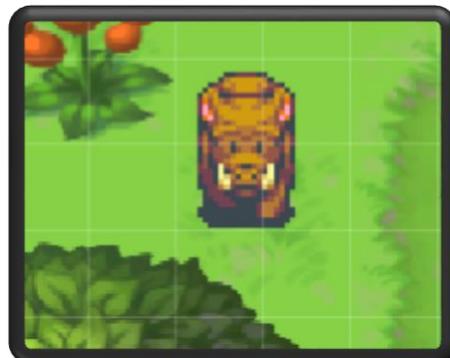


Ilustración 34. Objeto destructible animal

4.4.6 Sonido

El sonido es una parte crucial para generar una experiencia satisfactoria al jugador, ya sea por las propias melodías que acompañan los niveles, así como por los efectos que provocan un Feedback en el jugador a sus acciones.

Se han utilizado 39 pistas distintas repartidas entre música y efectos. Su implementación se ha basado en función de la localización del jugador, de la interacción con el entorno, o del movimiento que realizaba en ese momento.

4.4.7 HUD

El HUD diseñado ha intentado ser lo más simple posible. Informa al jugador de las vidas, las monedas recogidas la zona donde se encuentra, las llaves de las que dispone y en la parte superior de la puntuación obtenida hasta el momento.



Ilustración 36. HUD

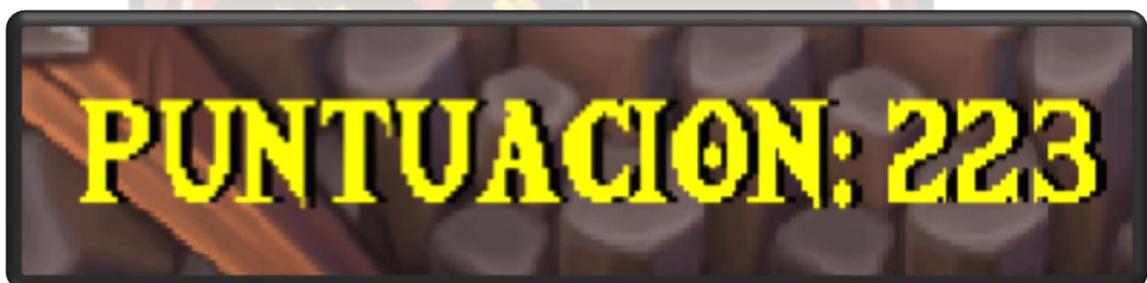


Ilustración 37. Puntuación

5. Diseño de niveles

5.1 Proceso de diseño

Para el diseño de niveles se ha seguido una metodología parecida para todos ellos. Para empezar, se deben incluir en el proyecto los scripts que compondrán las casillas de las rooms o mapas. El aspecto de las rooms inicial es el siguiente:



Ilustración 38. Estado inicial de una room

Cada casilla se representará adecuadamente en el momento en el que se definen, los tamaños del conjunto de casillas. Para explicar esto de forma directa se expone el siguiente ejemplo.

Se define una habitación con el siguiente tamaño:

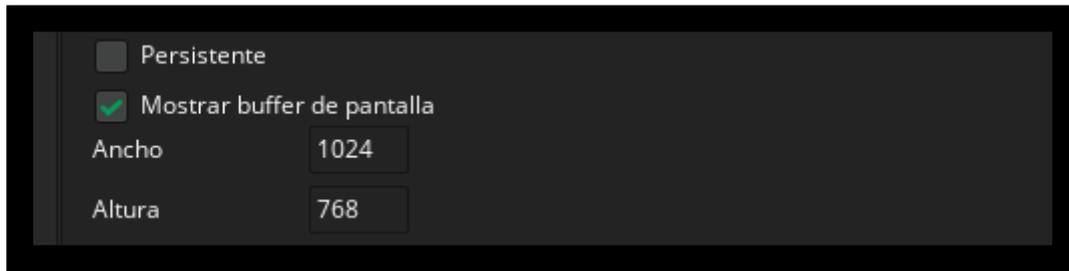


Ilustración 39. Configuración de tamaño de una room

Si por ejemplo escogemos un tamaño de casillas de 64x64. Tenemos que:

- **Ancho** = $1024/64 = 16$ Casillas de ancho
- **Alto** = $768/64 = 12$ Casillas de alto

La resolución o tamaño de estas casillas suele ser variable. De forma natural, una casilla de mayor tamaño o número de píxeles tendrá una definición mayor. En este caso se han utilizado casillas de 64x64 en todos los mapas salvo algunas excepciones puntuales.

Los sprites usados para el proyecto han sido adquiridos en el pack Humble Bundle RPG y son del artista Daniel Thomas.

5.1.1 Sprites

A continuación, podemos ver los sprites utilizados para cada uno de los conjuntos de casillas.

B
O
S
Q
U
E



Ilustración 40. Sprite de diseño de bosque



Ilustración 41. Sprite de diseño de ciudad



Ilustración 42. Sprite de diseño de interior

M I N A



Ilustración 43. Sprite de diseño de mina

Estos han sido los sprites utilizados para el diseño de todos los mapas que incluye nuestro juego. Adicionalmente, en el proyecto se incluían los de mazmorra/castillo, que se implementarán en versiones posteriores del mismo.

Una vez incluidos estos sprites en el proyecto como podemos observar:



Ilustración 44. Conjunto de sprites de mapa completo

5.1.2 Conjunto de casillas

Realizaremos el proceso de definir el encasillado con los sprites incluidos. Para ello es necesario definir un tamaño tanto horizontal como vertical. GameMaker realizará una división en función del tamaño que acabamos de definir, y generará el conjunto de casillas con el nombre que deseemos.

Como ejemplo incluimos el conjunto de casillas del script ciudad con el tamaño que comentamos previamente:

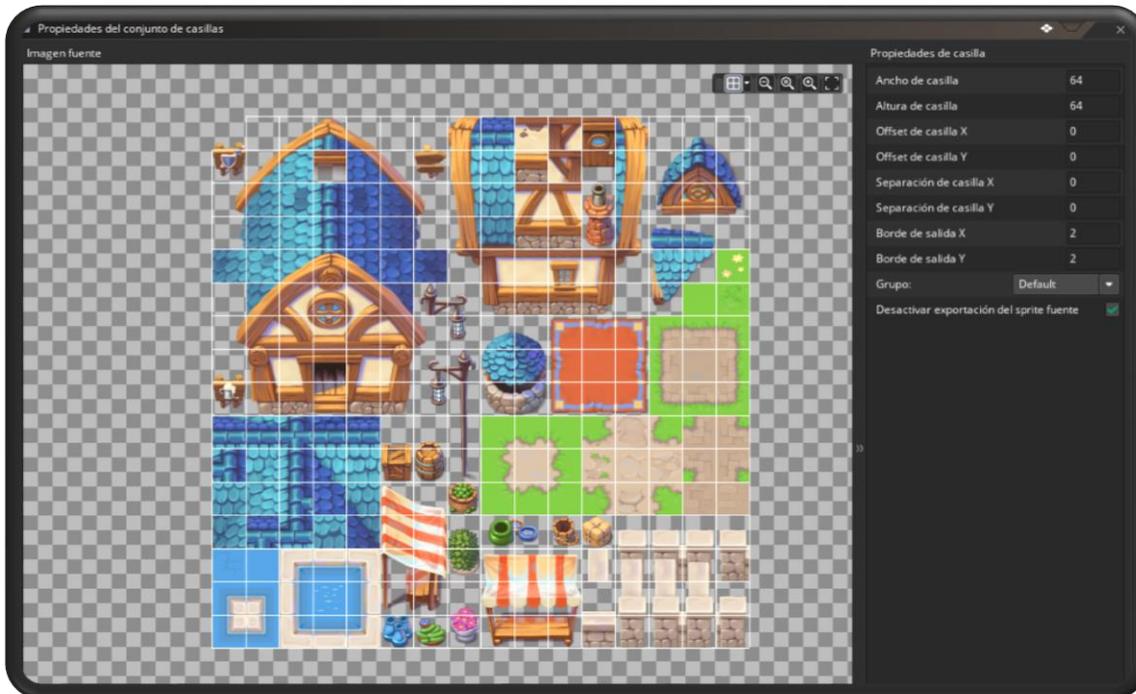


Ilustración 45. Definición del conjunto de casillas

5.1.3 Diseño de mapas

Con estos procesos se cuentan con las herramientas necesarias para incluir estas casillas en el mapa. Es necesario tener en cuenta la distribución en capas para los efectos de profundidad. Por lo tanto, se definirán tantas capas como sean necesarias y en ellas se distribuirán las casillas para generar la profundidad. Este es el diseño en capas del mapa Ciudad:

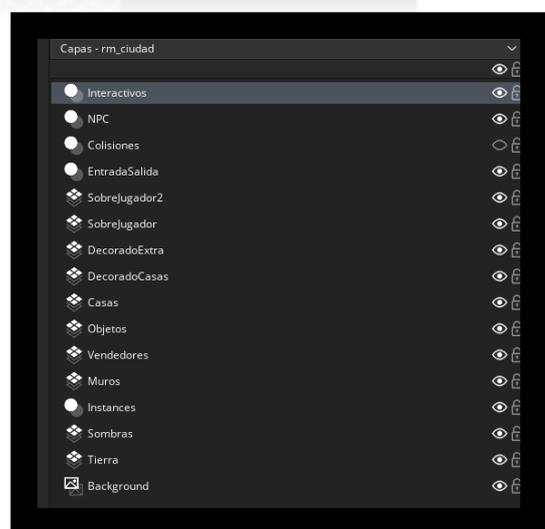


Ilustración 46. Capas del diseño del mapa Ciudad

A continuación, se incluyen todos los diseños de los mapas que podemos ver en el juego con las diferentes capas:

5.1.3.1 Mapa Cascada

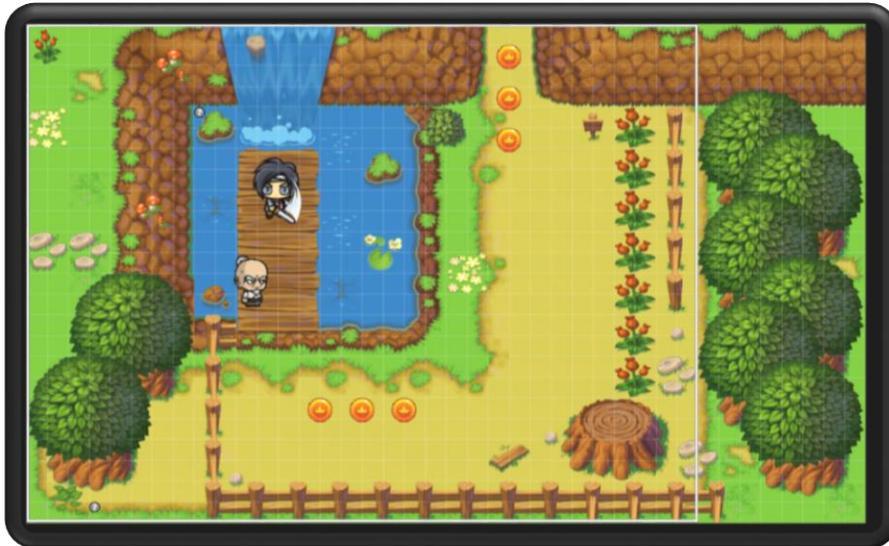


Ilustración 47. Mapa Cascada

5.1.3.2 Mapa Campo de Damaris

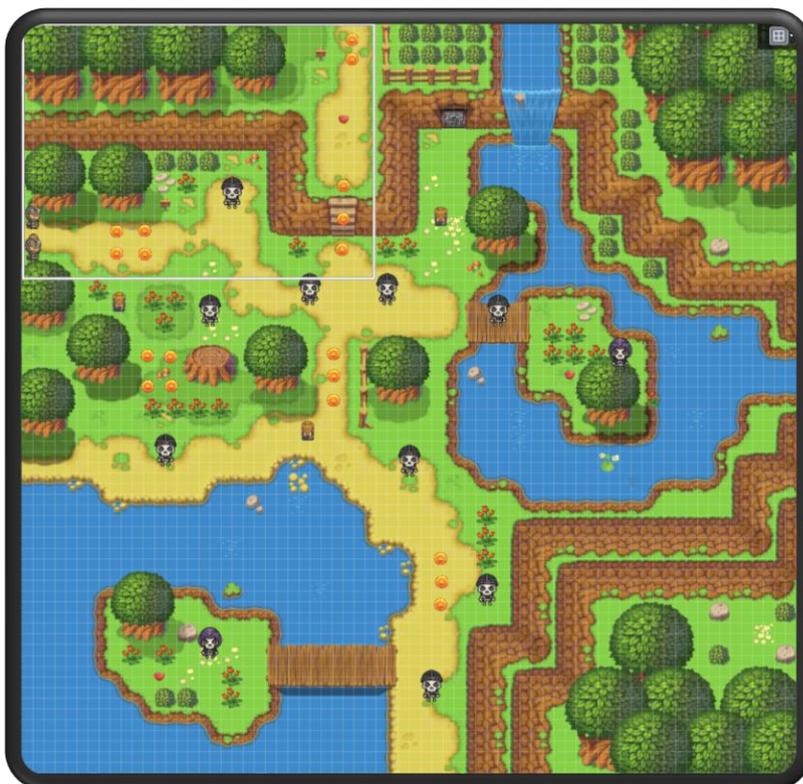


Ilustración 48. Mapa Campo de Damaris

5.1.3.3 Mapa Ciudad de Damaris



Ilustración 49. Mapa Ciudad de Damaris

5.1.3.4 Mapa de la taberna de Damaris



Ilustración 50. Mapa Taberna de Damaris

5.1.3.5 Mapa de la hermandad de aventureros



Ilustración 51. Mapa Hermandad de aventureros

5.1.3.6 Mapas de la mina de Damaris



Ilustración 52. Mapas de la mina

5.1.4 Elementos necesarios del diseño de niveles

Para entender los criterios que se han seguido para el diseño de niveles es necesario conocer los diferentes ítems necesarios para el diseño del mapa. Además de las propias casillas mencionadas anteriormente son necesarios una serie de controladores que darán vida al mapeado y prevendrán al jugador de realizar acciones no deseadas por parte del equipo de desarrollo.

Como se ha mencionado las casillas o sprites no dejan de ser imágenes que, repartidas en un lienzo, conforman un mapeado. Este lienzo puede estar dividido en multitud de capas y profundidades, pero, de forma lógica, no significan nada para el juego. Una caída, no es más que la imagen que precede a la del borde del precipicio.

Aunque algunos de estos objetos ya fueron mencionados en la fase técnica, es necesario comentarlos de nuevo para ver su repercusión en el mapeado para, a posteriori, ver un diseño de nivel completo y las decisiones tomadas:

- **Objetos interactivos:** Comparten las características del objetivo interactivo padre, pero tienen una implementación y objetivos distintos dependiendo del que haya sido incluido. A continuación, vamos a incluir la lista de objetos interactivos y sus funcionalidades.

Carteles

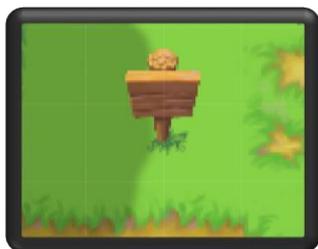


Ilustración 54. Objeto interactivo cartel 1

Los carteles permiten la interacción y nos muestran información relevante a la zona donde nos encontramos.



Ilustración 53. Objeto interactivo cartel 2

Cofres



Ilustración 55. Objeto interactivo cofre 1

Los cofres los encontraremos repartidos por el mundo, principalmente en las mazmorras. Incluyen objetos de bonificación y las llaves de las mazmorras.



Ilustración 56. Objeto interactivo cofre 2

Puertas con cerradura



Ilustración 58. Objeto interactivo puerta 1

Las puertas con cerradura requerirán de la llave apropiada para entrar. Nos mostrarán un cartel informativo indicándonos que llave nos falta y, en caso de tenerla nos permitirán el paso.

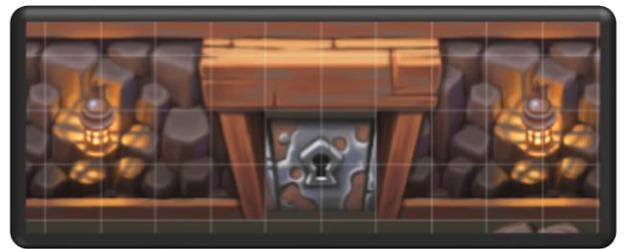


Ilustración 57. Objeto interactivo puerta 2

Monedas



Ilustración 60. Objeto interactivo moneda 1

Las monedas han sido animadas y se pueden recoger para aumentar los puntos del jugador. Posteriormente, una vez desbloqueada la tienda de la ciudad de Damaris, servirán para comprar mejores objetos.



Ilustración 59. Objeto interactivo moneda 2

Corazones



Ilustración 61. Objeto interactivo corazón 1

Los corazones ayudan al jugador a recuperar sus puntos de vida, cada uno de ellos recupera un punto de vida hasta un máximo de 3.



Ilustración 62. Objeto interactivo corazón 2

Flores



Ilustración 63. Objeto interactivo flores 1

Las flores son un objeto interactivo destructible que podrá conceder un corazón o una moneda.

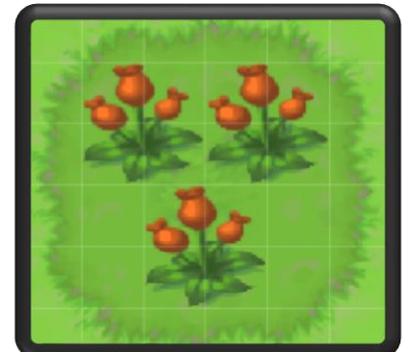


Ilustración 64. Objeto interactivo flores 2

Animales



Ilustración 66. Objeto interactivo animales 1

Del mismo modo que las flores, son un objeto destructible que nos puede dar un corazón, pero con una mayor posibilidad que otros objetos.



Ilustración 65. Objeto interactivo animales 2

NPC's

Existen más de 20 NPC's (Non Playable Character) distintos solo en el primer nivel con sus respectivas animaciones. Cada uno de ellos será interactuable y tendrá sus propias líneas de diálogo. Se han realizado diseños con sprites por separado de sus



Ilustración 68. Objeto interactivo NPC Rey

rostros para incluirlos en la conversación.

Estos personajes

nos darán información importante y, en ocasiones, serán clave para avanzar en la aventura.



Ilustración 67. Objeto interactivo NPC's

- **Enemigos:** Existen varios tipos de enemigos con sus propias características. Estos, estarán repartidos por el mapa y será necesario golpearles con nuestra espada para acabar con ellos. Todos tienen una probabilidad de soltar una moneda/corazón.

Soldado esqueleto



Ilustración 69. Enemigo soldado esqueleto

El enemigo más básico. Tiene la menor cantidad de puntos de vida y el ataque más lento.

Guerrero esqueleto



Ilustración 70. Enemigo guerrero esqueleto

Un enemigo más avanzado. Tiene más puntos de vida que el soldado esqueleto y un ataque más rápido.

- **Colisiones:** Tal y como se comentaba anteriormente, es necesario definir un objeto colisión para marcar los márgenes de los mapas. Este objeto simplemente reduce las velocidades del jugador para que no lo traspase. Una vez definido lo tendremos que colocar por el mapa para conformar el espacio de juego del jugador y separarlo del espacio no jugable.

En la siguiente ilustración tenemos un ejemplo de márgenes de un nivel definidos con colisiones.

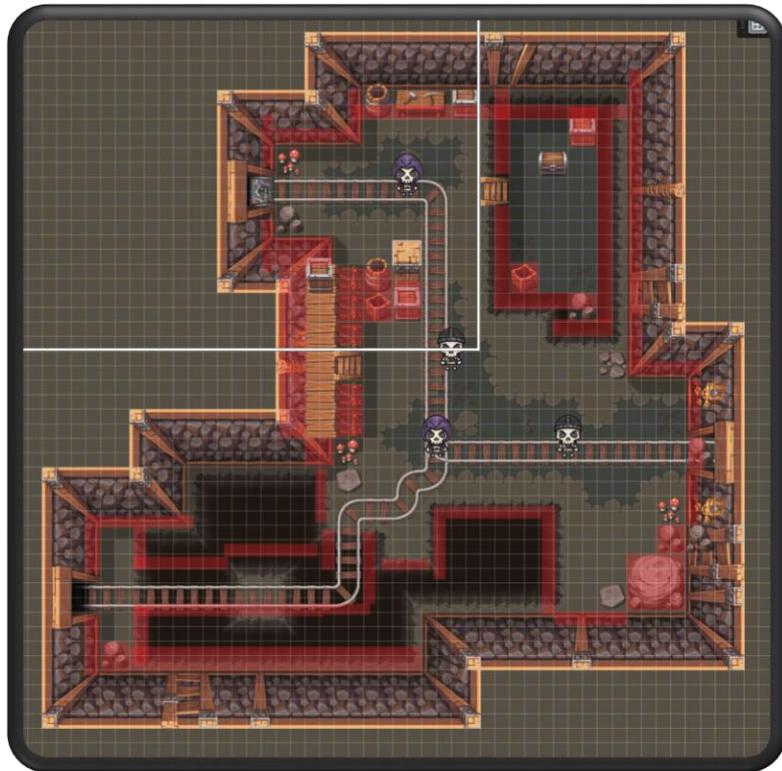


Ilustración 71. Colisiones en el mapa de la mina

- **Eventos:** Los eventos son un objeto parecido al de colisión que activan un evento específico. Estos eventos pueden ser una conversación o la entrega de un ítem clave, por ejemplo.
- **Entrada/Salida:** De la misma forma que los objetos evento, los entrada/salida se colocan en el mapa para cambiar entre los distintos mapas del mundo de Damaris.

Estos son la mayoría de los objetos que junto con las casillas conforman un nivel.

A continuación, vamos a comentar uno de los niveles del juego con todos los objetos incluidos que acaban de ser comentados.

El mapa de mayor tamaño del primer nivel es el que además incluye la mayor cantidad de ítems que se acaban de describir. Este es el mapa del campo de Damaris:



Ilustración 72. Diseño completo del mapa Campo de Damaris

La zona tiene cuatro puntos clave de conexión con otras zonas.

- 1** **Conexión con la zona de la Cascada.** El mapa inicial donde empieza la aventura y que da acceso a la zona del campo de Damaris
- 2** **Conexión con la ciudad de Damaris.** Camino que conecta el campo de Damaris con la ciudad.
- 3** **Conexión con el camino al castillo de Algart.** Zona deshabilitada en el primer nivel y que contiene 2 Npc's que guardan la entrada e impiden al jugador pasar.
- 4** **Puerta de acceso a la mina.** Esta puerta inaccesible en un primer momento se podrá abrir tras obtener la llave del jefe de la hermandad en la ciudad de Damaris, y con ella tendremos acceso a la mina.
- 5** **Monedas/Corazones.** Existen tanto corazones como monedas repartidos por el nivel. Las monedas de este nivel sirven además para guiar al jugador hasta la ciudad
- 6** **Plantas destructibles/Animales.** Ambos pueden ser destruidos permanentemente y recompensarán al jugador con corazones o monedas.
- 7** **NPC's.** Dos soldados nos impiden el paso al camino del castillo de Algart. Nos darán algo de información sobre lo que ocurre.
- 8** **Carteles.** Dos carteles nos dan la información necesaria de las zonas adyacentes al mapa actual.

Camino hacia la Ciudad



El camino que debe realizar el jugador para llegar a la ciudad de Damaris y obtener la llave que permite abrir la mina de Damaris.

Camino hasta la mina



El camino que debe realizar el jugador para llegar a la mina desde la ciudad de Damaris.

Todos estos elementos conforman el diseño de uno de nuestros niveles. La distribución de los mismos queda a disposición del diseñador de niveles que tiene que intentar que el jugador se sienta cómodo en los mismos y que en función del avance del juego resulten un desafío.

En este caso se encuentra uno de los mapas iniciales del juego que, además, sirven de interconexión entre varias zonas. Es por esto que encontramos una zona de intersección central que conectará estas zonas. Además de ello, se ha intentado seguir la lógica de agua comenzada en el mapa de la cascada, por lo que encontramos abundantes zonas con ella.

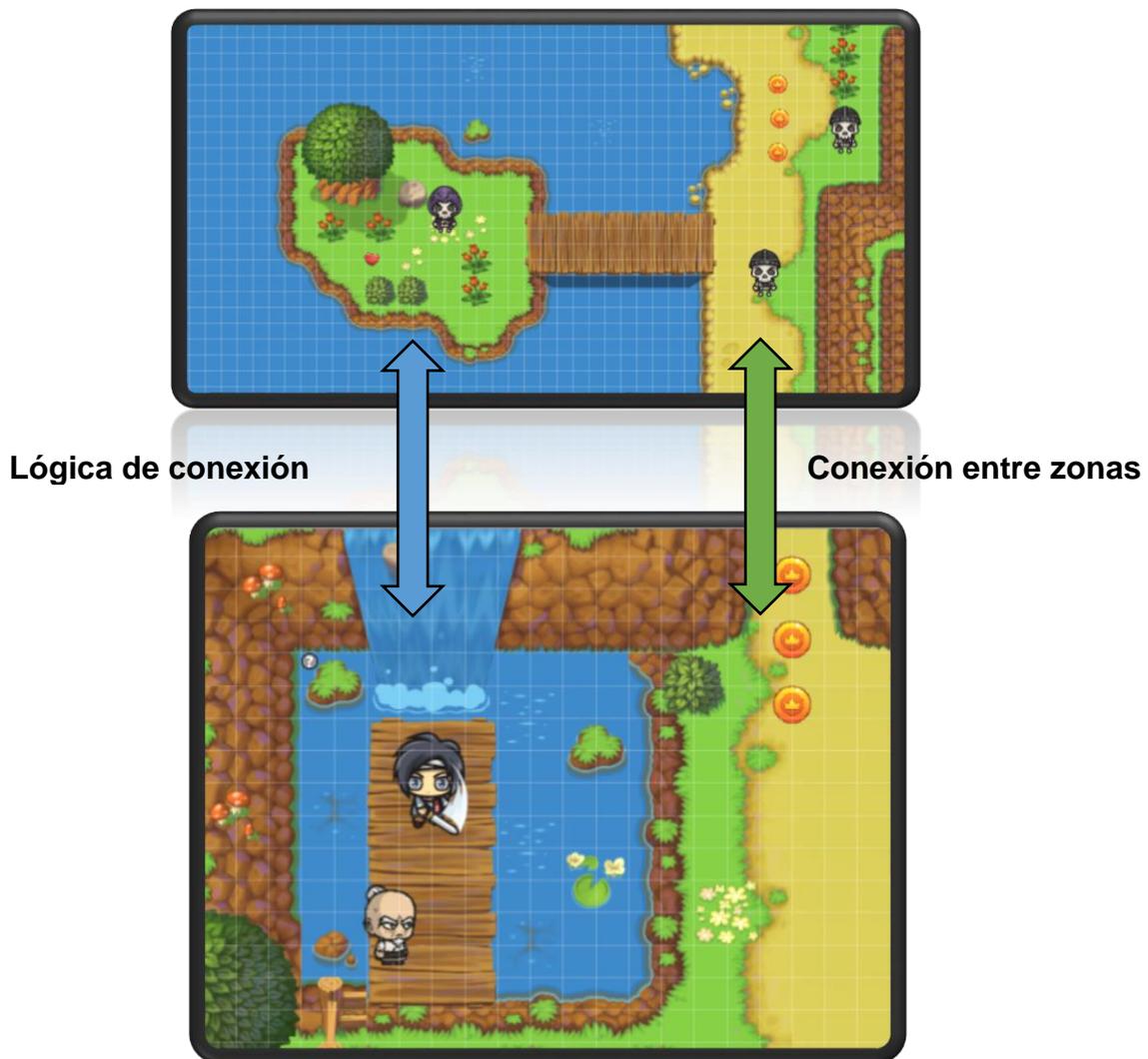


Ilustración 73. Lógica de diseño entre zonas

6. Manual de usuario

6.1 Requisitos del juego

Dado que el producto que vemos tiene unas características básicas no es necesario un hardware o software potente para hacerlo funcionar correctamente. En cualquier caso, dados los sprites utilizados y estudiando su consumo en memoria se plantearán unos requisitos que podrían ajustarse a Pc's incluso más antiguos.

MÍNIMO:

SO: XP

Procesador: Core 2 Duo

Memoria: 2 GB de RAM

Gráficos: Tarjeta de video discreta o integrada

Almacenamiento: 400 MB de espacio disponible

Tarjeta de sonido: Si

RECOMENDADO:

SO: Windows 8 / 7 / Vista / XP

Procesador: 2.4 GHz Quad Core 2.0 (or higher)

Memoria: 8 GB de RAM

Gráficos: Intel HD Graphics 4000 o superior, ATI Radeon HD-Series 4650 y superior, Nvidia GeForce 2xx-Series y superior.

Almacenamiento: 400 MB de espacio disponible

Tarjeta de sonido: Yes

6.2 Instrucciones del juego

Los controles serán mostrados al jugador al inicio del juego y pondrán encontrarse en la pantalla de inicio del mismo.



Ilustración 74. Controles del juego

7. Conclusiones

7.1 Conclusiones del trabajo

Finalizando este proyecto tengo sensaciones encontradas. La decisión de rehacer el diseño desde 0 fue acertada ya que ha permitido incluir las funcionalidades y mecánicas que se deseaban. Sin embargo, ha alterado de forma drástica los objetivos temporales marcados inicialmente. Por otro lado, no todas las funcionalidades pensadas desde un primer momento han sido incluidas.

El conocimiento adquirido de, no solo GameMaker, sino del desarrollo de (Elisabeth Casillas, 2013) videojuegos en general, ha sido sin duda lo mejor de este trabajo. Creo que la inexperiencia ha jugado una mala pasada, ya que en un primer acercamiento la dimensión que se le dio al proyecto era demasiado grande para los conocimientos y el tiempo del que disponía. Con todo esto, me siento muy satisfecho del trabajo realizado. La sensación de plantear un objetivo que tenía en mente como: *“Me gustaría que al entrar por la puerta iniciase una conversación, y a posteriori, comience una batalla”*. Desarrollarla con los elementos a mi disposición y, finalmente, reproducirla ha sido muy satisfactorio. Me atrevería a decir que nunca he disfrutado tanto en un desarrollo. El único lamento al respecto ha sido no haber realizado un enfoque más realista al comienzo del diseño y haber estado lastrado por las malas decisiones en este momento.

En conclusión, estoy muy satisfecho con los resultados obtenidos. Ver a nuestro héroe siendo una simple imagen y finalmente pelear desde la cascada hasta la mina ha sido fantástico.

7.2 Objetivos planteados. Reflexión crítica

Al llegar a la conclusión podemos decir que se han cumplido los objetivos planteados al inicio, pero no de la forma que nos gustaría. Desde luego, han existido tres motivos principales que han generado la mayor parte de los problemas:

La plataforma escogida: GameMaker me parece una plataforma maravillosa, pero ha sido un error escogerla. Es cierto que muchos desarrolladores la recomiendan para iniciados en el diseño de videojuegos en gran medida debido a su sistema DnD (Drag and Drop). Sin embargo, esto ha sido un arma de doble filo. Desde el principio de su uso descartamos utilizar DnD debido a que nos encontramos ante un trabajo de desarrollo personal. De nada sirve utilizar una plataforma que hace parte del trabajo del desarrollador si el objetivo principal es demostrar que es capaz de realizar un proyecto por sí mismo.

No solo esto era un problema. Como hemos comentado a lo largo de esta memoria, este tipo de desarrollos necesita un equipo multidisciplinar, el cual no tenemos. Por tanto, elementos tan importantes como los sprites de los personajes deben ser o bien diseñados, o adquiridos de forma externa. GameMaker cuenta con un mercado bastante pobre en ese sentido y, otras plataformas como Unity cuentan con diseños exclusivos para esa plataforma que hubiesen sido de gran ayuda y, de los cuales no podíamos disponer.

El tipo de juego escogido: Tras diseñar mapas, colisiones, eventos, vistas, etc., creo que un desarrollo de scroll lateral hubiese sido una opción más acertada. La posibilidad de centrar esfuerzos en el código propio o la implementación de mecánicas del juego, hubiese creado un producto más rico o, al menos, hubiese dado la impresión de más trabajado. El tipo escogido requiere de una gran parte de trabajo dedicado al diseño de mapas y de personajes. Este tiempo resta, en lo que a mecánicas jugables se refiere, y, puede dar la impresión de que es un producto simple, pero tiene tanto trabajo o más que otros enfoques.

El último motivo lo comentaremos en el siguiente punto.

7.3 Seguimiento de la planificación y metodología

El seguimiento comenzó de manera correcta realizando un primer acercamiento al diseño propuesto. Sin embargo, el desarrollo realizado no se correspondía con los objetivos planteados, ya que generaba unas mecánicas para la narrativa e interacción con otros personajes buena, pero nulas en cuanto al combate se refiere. A esto hay que sumarle el hecho de disponer de muy pocos recursos a nivel de sprites, que generaban limitaciones en las mecánicas a implementar.

Como consecuencia me vi obligado a reiniciar el proyecto a mitad de su desarrollo lo que trastocó por completo la planificación del proyecto. Además de esto, apenas se pudo reutilizar el código implementado por lo que partimos de 0.

Todos estos errores vienen de la inexperiencia en primer punto y del mal enfoque realizado al inicio. Ha sido necesario recortar habilidades al personaje que, en principio llevaría espada y escudo y un sistema de inventario para ir equipando diferentes ítems.

La metodología se ha basado en Estudio → Aplicación → Corrección, y en este sentido ha sido bastante exitosa, teniendo en cuenta la nula experiencia y conocimientos de la plataforma usada.

Podemos concluir en este punto que, se ha seguido una planificación de forma adecuada, pero esta ha sido alterada debido a los problemas, sin embargo, es necesario añadir que la planificación realizada tras el cambio de los cimientos del proyecto ha sido exitosa por lo que podemos afirmar que hemos podido corregir un error grave en la propia gestión del proyecto.

7.4 Líneas de trabajo futuro

Tengo tantas ideas que me gustaría implementar en el mundo de los videojuegos que por una parte pienso que el camino de My First Hero acaba en esta entrega. Me gustaría utilizar los conocimientos adquiridos en realizar un proyecto desde otro enfoque. Por otro lado, siendo como ha sido mi primer producto de este tipo lo siento como un hijo y, me gustaría continuar. Posiblemente en este sentido reiniciaría parte de lo diseñado para implementar todas las mecánicas que inicialmente se plantearon.

Es cierto que los desarrollos tipo Metroidvania con un enfoque Roguelike son el modelo que me llama más la atención actualmente. Sé que los algoritmos usados en estos proyectos son muchísimo más complejos y que requieren de un desarrollo lógico excepcional y, es por esto, que me gustaría adentrarme en ellos, me gusta el reto que plantean.

No descarto trabajar en este proyecto en el futuro puesto que creo que revisado y, con un mayor número de recursos, tiene gran potencial. Lo que es seguro es que este no será mi último desarrollo de videojuegos, My First Hero ha sido el acercamiento necesario para explorar un mundo apasionante para mí.

Para cerrar mi trabajo dejaré me gustaría dejar mi impronta con una pequeña reflexión:

“Los videojuegos fueron mi puerta de entrada a esta etapa desde que fui pequeño. Resulta paradójico que también sean la puerta de salida. El círculo se cierra.”

8. Glosario

- **EDSAC:** Acrónimo de Electronic Delay Storage Automatic Calculator, una de las primeras computadoras creadas de origen británico.
- **ATARI:** una de las productoras de videojuegos independientes más grande de EEUU y principales fabricantes de consolas en el mercado.
- **TIC:** Tecnología de la Información y comunicación
- **SPRITE:** una imagen de mapa de bits gestionada por un hardware especializado usado para los protagonistas de los videojuegos u otros personajes
- **DIAGRAMA GANTT:** herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades
- **CELL SHADING:** en español, sombreado plano. Es un tipo de renderización diseñada para hacer que los gráficos por computadora parezcan dibujadas a mano.
- **MMORPG:** acrónimo en inglés de Videojuegos de rol multijugador masivos en línea. Por ejemplo, World of Warcraft (WOW)
- **TRIPLE A:** clasificación informal utilizada para los videojuegos que son producidos y distribuidos por una distribuidora importante o editor importante.
- **DIRECTX 11:** es una interfaz de programación de aplicaciones desarrollada para facilitar las complejas tareas relacionadas por la programación de juegos y videos en Windows
- **STEAM:** plataforma de distribución digital, gestión digital de derechos, comunicaciones y servicios multijugador desarrollada por Valve Corporation.
- **SO:** acrónimo de sistema operativo
- **HUD:** información que en todo momento se muestra en pantalla durante a partida, generalmente en forma de iconos y números. Por ejemplo, puntos de vida.
- **NPC:** acrónimo en inglés de personaje no jugable.

9. Bibliografía

- Cosmonaut, F. (s.f.). *Tutoriales sobre GameMaker*. Obtenido de <https://www.youtube.com/channel/UCKCKHxkH8zqV9ltWZw0JFig>
- DEV. (s.f.). *Libro blanco de desarrollo de videojuegos*. Obtenido de <http://dev.org.es>
- Elisabeth Casillas, S. G. (2013). Juegos Indies Nacionales. Al asalto del PC. *Revista Cactus*.
- Kent, S. L. (2001). *La gran historia de los videojuegos*. Watcher.
- Nintendo. (2017). Obtenido de <http://es.nintendo.wikia.com>
- Profe, H. (s.f.). *Tutoriales RPG GameMaker*. Obtenido de <https://www.youtube.com/channel/UCtjAOyZmqDXO-Oz87cZnWgw>
- silicio, G. d. (2007). *Historia de los videojuegos*. Obtenido de <http://historia-videojuegos.blogspot.com/2007/12/OXO.html>
- Simone Belli, C. L. (2008). Breve historia de los videojuegos. *Athenea Digital*, 14, 159-179.
- Vara, C. F. (2009). *The tribulations of adventure games: integrating story into simulation through performance*. Atlanta: Institute of Technology.
- YoYoGames. (s.f.). Obtenido de Página Web de GameMaker: <https://www.yoyogames.com/gamemaker>