TFG EN INGENIERÍA INFORMÁTICA, COLEGIO DE INGENIERÍA (EE), UNIVERSIDAD AUTÓNOMA DE BARCELONA (UAB)

Dungeon Diver: Desarrollo de un videojuego Hack and Slash en 2D

Javier García Cantero

Resumen-- La industria del videojuego genera al año billones de dólares en beneficios y crea muchas oportunidades de trabajo. Por ese motivo he decidido crear Dungeon Diver, un videojuego de acción en 2D, donde el objetivo del jugador es llegar al fondo de la mazmorra, sobreviviendo a los ataques de enemigos que están por el camino y posteriormente, derrotar al enemigo final. El proyecto se ha realizado con el motor Unity, que usa C# como lenguaje para el scripting que da a los personajes comportamiento inteligente, y siguiendo una metodología de desarrollo ágil. El objetivo de este proyecto es crear una versión jugable de principio a fin, trabajando en todas la fases del desarrollo de un videojuego.
Palabras clave-- Videojuego, Unity, C#, Inteligencia Artificial, Scripting, ágil.

Abstract-- The videogame industry generates billions in revenue every year and creates lots of job opportunities. Because of that I decided to create Dungeon Diver, a 2D action video game, where the player must reach the end of the dungeon, surviving enemy attacks along the way and defeat the final enemy. The entire project has been made using the Unity Engine, which uses C# as scripting language to give the characters intelligent behavior. I followed an agile methodology. The objective of this project is to create a versión of the game that is playable from beginning to end, working on every stage of video game development.

Index terms-- Video game, Unity, C#, Artificial Intelligence, Scripting, agile.

1 Introducción

El objetivo de este trabajo de final de grado consiste en crear un videojuego utilizando el motor[1] UNITY, haciendo uso de C# para crear los scripts[2] y dotar a los enemigos de comportamiento inteligente, y la herramienta Aseprite para dibujar los sprites del juego.

El juego consiste en llegar superar niveles hasta llegar al enemigo final ("Boss") y derrotarlo. Para esto, el jugador dispone de número de habilidades que le ayudarán a superar los enemigos que encuentre por el camino y finalmente derrotar al enemigo "Boss".
Cada enemigo tiene un comportamiento diferente respecto a cómo interactúa con el jugador, e intentarán dañar al jugador hasta que este pierda todos sus puntos de vida.
La versión final tiene 6 enemigos: slime, mago, caballero, orco, mandíbula del caos y gazer. Cada uno de estos enemigos tiene un comportamiento específico que se explicará con detalle en la sección 5. La partida termina cuando el Boss es derrotado o cuando el jugador se queda sin puntos de vida, en cuyo caso volverá a aparecer al principio del nivel.

Este informe inicial está separado en 8 secciones:
La sección 1 contiene la introducción, donde se habla de la motivación, el contexto y se da una idea general del proyecto. La sección 2 enumera los objetivos que se han de cumplir para considerar el proyecto como finalizado. Se dividen en objetivos generales y específicos. La sección 3 muestra la metodología a seguir durante el desarrollo del proyecto. La sección 4 contiene los requisitos, divididos en funcionales y no funcionales. La sección 5 explica el funcionamiento del juego y todas sus partes. La sección 6 muestra resultados del proyecto. En la sección 7 encontramos la conclusión. La sección 8 contiene los agradecimientos. Por último, la sección 9 contiene la bibliografía.

1. 1 Motivación

Un futuro en la industria del videojuego fue una de las razones por las cuales decidí cursar la carrera de Ingeniería Informática. Los videojuegos son algo muy prevalente en mi vida desde una temprana edad y siempre he querido recoger todas las experiencias que he ido acumulando a lo largo de los años y utilizarlas para crear algo propio. Ahora que la carrera me ha dado los conocimientos para desarrollar mi propio videojuego, he querido usar el trabajo de final de grado como una oportunidad para dar el primer paso en lo que espero que sea un futuro profesional en la industria del videojuego. Además, el proceso de desarrollo de un videojuego desde cero pone a prueba muchos de los conocimientos que la mención de Ingeniería del Software imparte.

1.2 Contexto

La industria del videojuego se vuelve cada año más importante. Mueve al año billones de dólares y algunos juegos obtienen beneficios que se acercan o sobrepasan a los de las películas más taquilleras del año. Cada vez, más personas juegan de manera regular, quizás en un ordenador, en una consola o en un móvil [1]. Para muchas personas no es más que un pasatiempo, pero para otros es un trabajo, un campo al que dedicar su vida profesional, y al que aspirar durante los estudios.

El desarrollo de videojuegos es un campo muy amplio que requiere diferentes tipos de habilidades, desde conocimientos técnicos en lo que se refiere a la programación del videojuego y la gestión del proyecto, a aspectos más artísticos como el diseño del juego, tanto en el aspecto visual como en las mecánicas y sistemas del mismo.

2. Objetivos

2.1 Objetivos Generales

El objetivo principal del proyecto es llevar a cabo el desarrollo de una versión jugable de un videojuego utilizando Unity [3] y C#.

El juego ha de ser del género hack and slash, con combate en tiempo real. Además de desarrollar el videojuego en sí, se ha de construir un Game Design Document(GDD). Se trata de un documento que se usa en el desarrollo de videojuegos y que contiene la visión del mismo y sirve de guía durante el desarrollo.

[1]Serie de rutinas de programación que permiten el diseño, la creación y la representación de un videojuego.

[2] Programas creados con funcionalidades determinadas.

El juego ha de ser jugable de principio a fin sin errores que impidan que se complete la partida. En la figura 1 se puede ver un juego parecido al que se va a desarrollar.

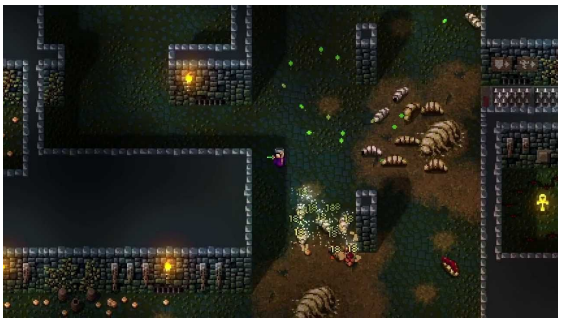


Figura 1. Hammerwatch, ejemplo de un videojuego similar al que se pretende desarrollar

2.2 Objetivos específicos

1. Construir un Game Design Document completo.
2. Crear un sistema de combate en tiempo real.
3. Crear dos niveles con enemigos normales[3] y un nivel con un enemigo más complejo o "Boss"[4].
4. Crear un elenco variado de enemigos normales.
5. Crear un repertorio de movimientos variados para el protagonista.
6. Crear una inteligencia artificial capaz de superar un nivel.
7. Crear un sistema de generación automática de niveles.

3. Metodología

3.1 Metodología de trabajo

Se ha utilizado una metodología scrum[1]. El trabajo se divide en sprints de 1 semana. Cada lunes se planifican los objetivos de la semana y el trabajo para cumplirlos. Por lo general, el sprint consiste en una fase de diseño, donde se planea como cumplir los objetivos, como por ejemplo, buscar información sobre un script que hay que implementar; y una fase de implementación, donde se efectúa el trabajo en sí. En la figura 2 se presenta un diagrama que explica esta metodología.

La naturaleza del desarrollo de videojuegos es muy cíclica, por lo que se está constantemente diseñando, implementando, testeando y vuelta de nuevo a diseñar. Si no se cumplen los objetivos de la semana, estos pasan a la siguiente.

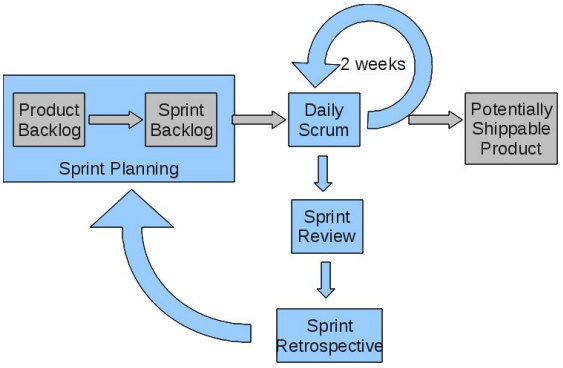


Figura 2. Diagrama de la metodología scrum. Paul Gestwicki

En cuanto a documentación, se utiliza un game design document o "GDD"[2], que se actualiza y edita constantemente, y que contiene la visión y las características del proyecto. La figura 3 muestra un ejemplo de la estructura de un GDD.
Se ha utilizado GitHub para el control de versiones.

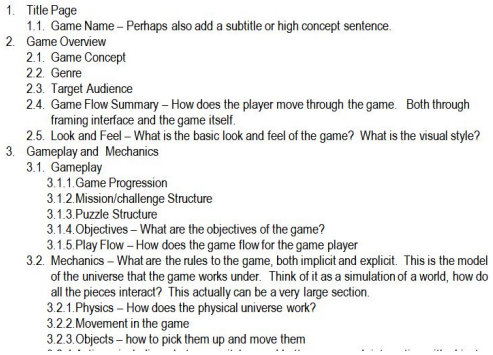
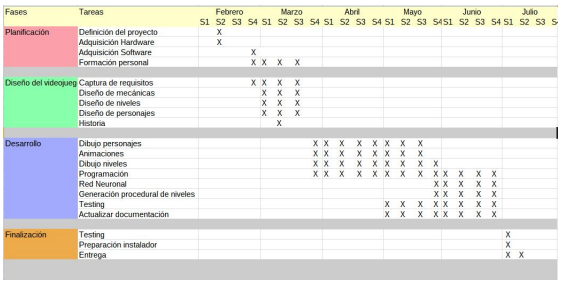

1. Title Page
   1.1. Game Name – Perhaps also add a subtitle or high concept sentence.
2. Game Overview
   2.1. Game Concept
   2.2. Genre
   2.3. Target Audience
   2.4. Game Flow Summary – How does the player move through the game. Both through framing interface and the game itself.
   2.5. Look and Feel – What is the basic look and feel of the game? What is the visual style?
3. Gameplay and Mechanics
   3.1. Gameplay
      3.1.1.Game Progression
      3.1.2.Mission/challenge Structure
      3.1.3.Puzzle Structure
      3.1.4.Objectives – What are the objectives of the game?
      3.1.5.Play Flow – How does the game flow for the game player
   3.2. Mechanics – What are the rules to the game, both implicit and explicit. This is the model of the universe that the game works under. Think of it as a simulation of a world, how do all the pieces interact? This actually can be a very large section.
      3.2.1.Physics – How does the physical universe work?
      3.2.2.Movement in the game
      3.2.3.Objects – how to pick them up and move them

Figura 3. Ejemplo de parte de la estructura de un game design document

[3] Se considera enemigo normal un enemigo que aparece constantemente y que no tiene mecánicas complejas.
[4] Se considera "Boss" un enemigo que aparece pocas veces o una sola vez y que tiene más movimientos y mecánicas de comportamiento que los enemigos normales.

3.2 Metodología de desarrollo



Fases	Tareas	Febrero				Marzo				Abril				Mayo				Junio				Julio		
		S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3
Planificación	Definición del proyecto		X																					
	Adquisición Hardware		X																					
	Adquisición Software				X																			
	Formación personal				X	X		X		X														
Diseño del videojueg	Captura de requisitos				X	X		X		X														
	Diseño de mecánicas					X		X		X														
	Diseño de niveles					X		X		X														
	Diseño de personajes					X		X		X														
	Historia					X																		
Desarrollo	Dibujo personajes									X	X		X		X		X							
	Animaciones									X	X		X		X		X							
	Dibujo niveles									X	X		X		X		X							
	Programación									X	X		X		X		X	X		X				
	Red Neuronal																	X		X				
	Generación procedural de niveles																	X		X				
	Testing														X		X	X		X				
	Actualizar documentación														X		X	X		X				
Finalización	Testing																					X		
	Preparación instalador																					X		
	Entrega																					X		

Figura 4. Diagrama de gantt con las fases del proyecto. Elaboración propia

La figura 4 muestra el cronograma del proyecto. La fase de planificación consta de la definición del proyecto, la adquisición de las herramientas y aprender a utilizarlas. Cabe recalcar que la formación probablemente seguirá durante todo el proyecto.
En la fase de diseño se plantea la historia, las mecánicas del juego, el diseño y la estética de los niveles, la interfaz y los personajes, tanto NPC[5] (non-player character), como enemigos y el protagonista.
En la fase de desarrollo se implementarán los diseños estéticos en el motor, se crearán las animaciones de los objetos y personajes y se programara la lógica del juego, siendo esta el comportamiento de los enemigos, objetos en el mapa, los movimientos del jugador, distintos eventos, etc. Durante toda esta fase se hace testing constantemente en el Software, siguiendo las directrices del *test driven development*.
Por último, en la fase de finalización se acabarán de testear las funcionalidades y se preparan los materiales de la entrega (Ejecutables, informes, documentación).

4. Requisitos

4.1 Requisitos funcionales

1. La perspectiva debe ser top-down en 2D.
2. El jugador debe poder moverse en 8 direcciones.
3. El jugador debe tener un mínimo de 4 movimientos o “habilidades” (Disparar, Bloquear, etc.).
4. Un mínimo de 3 niveles.
5. Un mínimo de 5 enemigos normales.
6. Un enemigo más complejo (Boss).
7. Crear un GDD completo

4.2 Requisitos no funcionales

1. Funcionar en Windows 10.
2. Utilizar el motor Unity y C#.
3. Tiempos de carga no superiores a 30 segundos.
4. Latencia de inputs menor a 500 ms.
5. Mínimo 30FPS (Imágenes por segundo) estables.
6. El sistema de combate debe ser en tiempo real.
7. Tener una inteligencia artificial que evoluciona mediante redes neuronales

5. Funcionamiento del juego

El juego es del género hack and slash[5], un género donde el aspecto más importante es el combate a tiempo real. Tendrá una perspectiva top-down en 2D.

Las interacciones físicas entre elementos del juego se llevan a cabo mediante colisionadores[6], que detectan si dos objetos entran en contacto. A partir de ese contacto se utilizan scripts para crear diferentes comportamientos[6]. En la figura 5 podemos ver el colisionador rodeando al jugador.

Figura 5. El cuadrado verde representa el colisionador

5.1 Diseño

Para dibujar los sprites del juego he utilizado el programa Aseprite[8], que está especialmente hecho para hacer dibujos con estética pixelada. La interfaz del software se puede ver en la figura 6.

[5] Personaje no controlado por el jugador.

[6] Objeto que se usa como referencia para detectar si dos cuerpos ocupan el mismo espacio

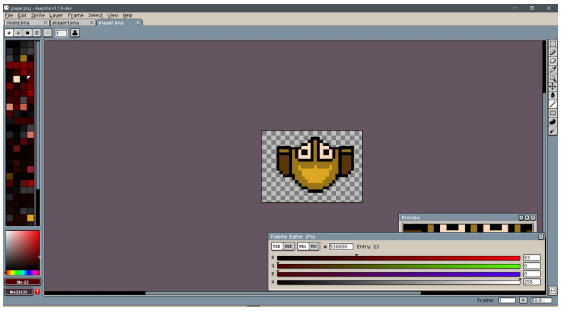


Figura 6. Interfaz de Aseprite

5.2 Mecánicas del Jugador

El jugador se puede mover en 8 direcciones (ver figura 7) y mirar en 360º, siendo la posición del ratón lo que determina la dirección en la que mira (y ataca) el personaje.

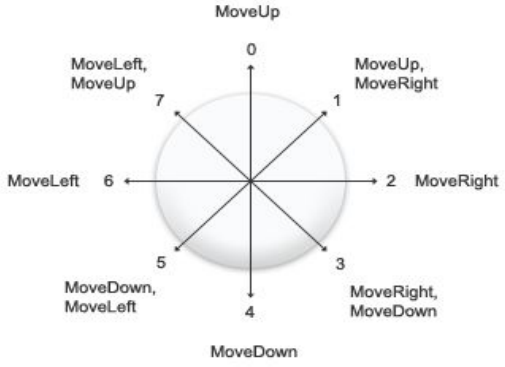


Figura 7. Direcciones en las que el jugador se puede mover

La cámara sigue al jugador y lo mantiene siempre centrado, en lugar de estar fija y moverse cuando el jugador llega al borde de la pantalla.
El diseño del jugador es un caballero con armadura dorada, que podemos ver en la figura 8.

Figura 8.Personaje que controlará el jugador

El jugador puede hacer un ataque a larga distancia, que podemos ver en la figura 9.

Figura 9. Jugador atacando a un enemigo con el ataque a distancia

También posee habilidades que el jugador puede utilizar para superar el juego, como un acelerón en una dirección y un movimiento de bloqueo, que se puede ver en la figura 10.

Figura 10. Jugador poniendo un escudo que bloquea proyectiles

La salud del jugador se mide en forma de un número que muestra los puntos de vida restantes. Además se podrá recuperar vida al recoger corazones del suelo durante la partida, que aparecen al derrotar enemigos. El diseño de estos corazones se presenta en la figura 11.

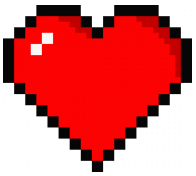


Figura 11. Corazón que cura al jugador

5.3 Mecánicas de enemigos.

Slime

Figura 12. Slime

En la figura 12 podemos ver el slime, el enemigo más básico del juego, cuyo comportamiento consiste en acercarse al jugador y tocarle. Si le toca, el jugador pierde un punto de vida.

Mago

Figura 13. Mago

En la figura 13 podemos ver el mago, un enemigo más complejo que el slime. Persigue al jugador pero a una distancia segura, alejándose si el jugador se acerca demasiado, y lanza proyectiles que dañan al jugador, que se pueden ver en la figura 14.

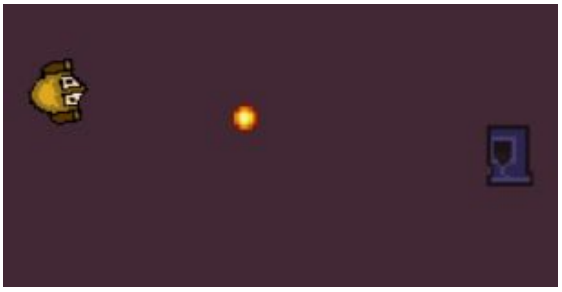


Figura 14. Mago atacando al jugador

Caballero

Figura 15 Caballero

En la figura 15 se presenta al caballero, un enemigo que patrulla, como se puede ver en la figura 16, y defiende una zona de los los ataques del jugador, atacando cuerpo a cuerpo.

Figura 15. Caballeros patrullando la zona de la llave.

Orco

Figura 17. Orco

En la figura 17 vemos al Orco, que carga en línea recta al ver al jugador, y que resiste más al daño que otros tipos de enemigos.

Mandíbula del caos

Figura 18.Mandíbula del caos

En la figura 18 se presenta la mandíbula del caos. Este enemigo ataca al jugador lanzando proyectiles

mientras se aleja. Por donde camina va cubriendo el terreno de ácido que daña al jugador si lo pisa. Este ácido se puede ver en la figura 19,

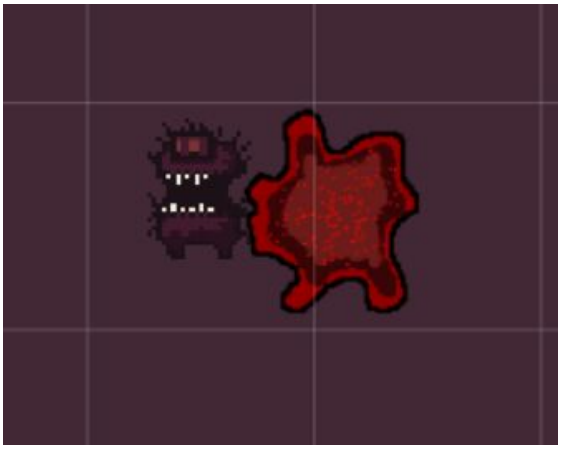


Figura 19. Ácido creado por el enemigo

Gazer

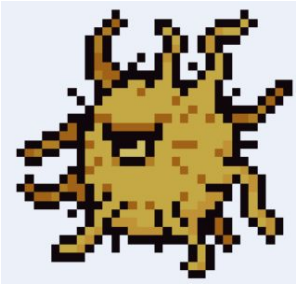


Figura 20.Gazer

En la figura 20 vemos al enemigo final del juego, con más movimientos que el resto. Dispara proyectiles en línea recta o en abanico, crea zonas que dañan al jugador si las pisa y ataca cuerpo a cuerpo.

5.4 Niveles

El jugador avanzará por los dos primeros niveles en busca de una llave, que podemos ver en la figura 21, que utilizará para abrir la puerta al siguiente nivel.

Figura 21. LLave

Los niveles están repletos de enemigos que intentan impedir el avance del jugador. En el nivel final, el jugador ha de derrotar al enemigo final “Gazer”. Si lo consigue, habrá ganado la partida.

Al diseñar los niveles se han utilizado grandes habitaciones donde hacer que el jugador se enfrente a un gran número de enemigos y pasillos en los que encontrará menos resistencia. En ambos hay una bifurcación que lleva a la puerta al siguiente nivel o a la llave para abrir la puerta. Esto se puede ver en las figuras 22 y 23. Se ha decidido que el jugador tenga que volver por el camino que ha venido al recoger la llave para que se vuelva a enfrentar a los enemigos que no haya derrotado, en el caso de que simplemente haya intentado huir. El tercer nivel, que se puede ver en la figura 24, donde el jugador se enfrenta al “Gazer” es una gran habitación cuadrada con amplio espacio para esquivar.

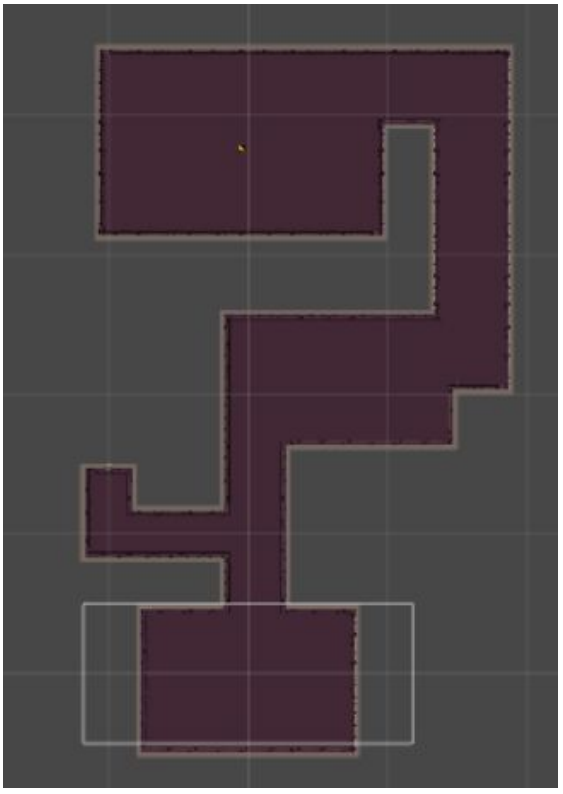


Figura 22.Primer nivel, sin enemigos

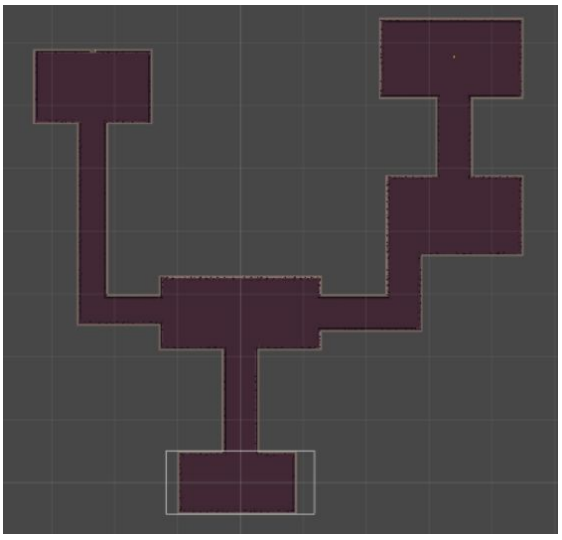


Figura 23.Segundo nivel, sin enemigos

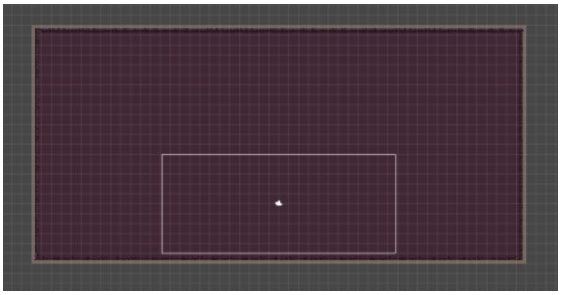


Figura 24.Tercer nivel, sin enemigos

Para “pintar” los niveles se ha utilizado la paleta de tiles[7], presentada en la figura 25, que permite añadir texturas al grid[8] cuadro a cuadro [7].

Figura 25.Paleta de tiles

[7] Unidad en la que se dividen los dibujos que se usan para crear el nivel.

[8] Cuadrícula sobre la que se diseña el nivel.

Las paredes y el suelo del nivel se han creado como dos objetos distintos, para hacer más sencilla la detección de colisiones, como se puede ver en la figura 26.

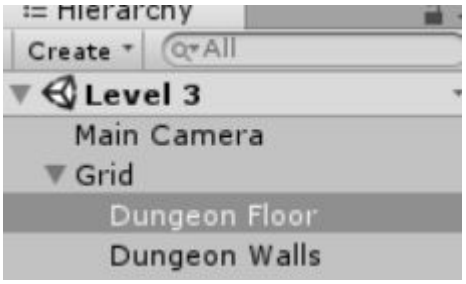


Figura 26.Objetos en la escena[9]

Al tener las paredes en un objeto a parte, simplemente al añadir “Tilemap collider 2D”, que se presenta en la figura 26, se añaden las cajas de colisión automáticamente, como podemos ver en la figura 27.

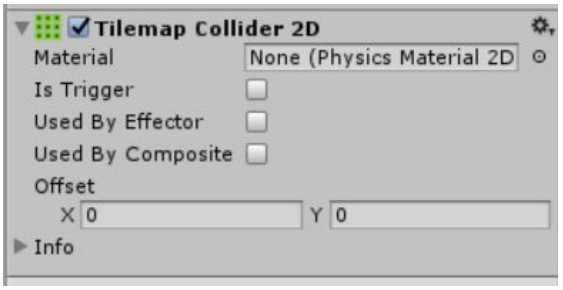


Figura 26.Atributos del colisionador 2D

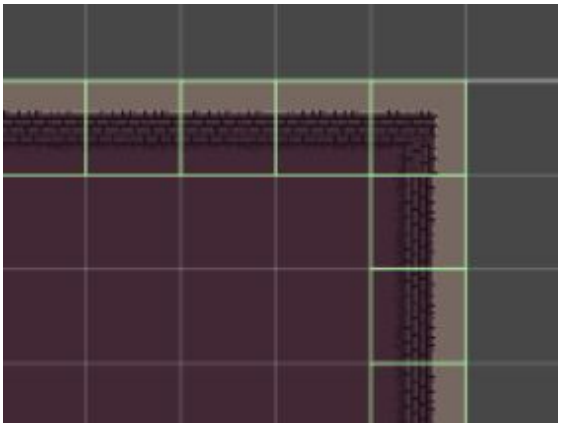


Figura 27. Colisionador en el nivel

5.5 Segundo modo de juego

El segundo modo de juego, el modo horda, donde el jugador ha de aguantar todo el tiempo posible contra oleadas infinitas de enemigos. En la figura

[9] Partes en las que el motor UNITY divide cada nivel del juego

28 se pueden ver los puntos de aparición de los enemigos.

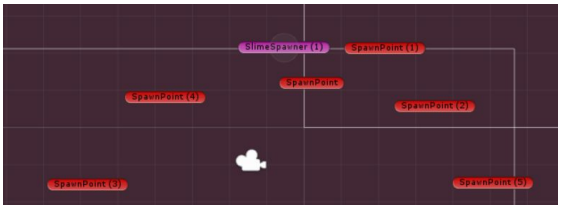


Figura 28. Puntos de aparición de enemigos

5.6 El menú principal

Desde el menú se puede acceder a los dos modos de juego y salir de la aplicación, pulsando los botones que vemos en la figura 29. Se puede volver a este menú mientras jugamos si pulsamos la tecla ESC para pausar el juego, y luego pulsamos la tecla M.

Figura 29.El menú del juego

6 Resultados

6.1 Estado del proyecto.

Si bien el juego puede ser completado de principio a fin, no se encuentra en un estado en el que se pueda comercializar. En un futuro se quiere implementar una IA con redes neuronales y se necesita más feedback de distintos usuarios para mejorar la calidad y crear así, un juego más divertido. Se puede decir que el juego está en un estado de Alfa[10]. Cabe destacar que hoy en día se lanzan juegos en estados muy tempranos para dar una oportunidad a los usuarios de formar parte del desarrollo.

6.2 Dificultades encontradas.

En las primeras etapas del desarrollo se encontró que Unity no estaba configurado por defecto para trabajar en 2D desde una perspectiva top-down. En la configuración por defecto la gravedad actúa en el eje Y, en lugar de en el eje Z, que es el que determina la altura en la perspectiva en la que trabajo, lo que hace que debido a la gravedad, por defecto, los personajes se “caigan” hasta la parte inferior de la pantalla. Para solventar este problema lo primero fue hacer que la gravedad no afecte a los personajes, pero luego se descubrió una opción para cambiar el eje sobre el que actúa la gravedad.

A finales del desarrollo se tuvo que cambiar cómo funcionaba el movimiento de todos los personajes del juego. Inicialmente se había hecho modificando el “transform” de los objetos, que es donde se almacena la posición del objeto en el espacio. Sin embargo, durante el testing de versiones más avanzadas, esto resultaba en comportamientos irregulares cuando el objeto interactuaba con un colisionador, ya que al modificar el transform era como si el objeto se teletransporta muchas veces en distancias muy cortas. Para solucionar esto, se cambió el script para que al moverse se añadiera un vector de velocidad en la dirección deseada. Esto causó algunos efectos indeseados como que los personajes mantuvieran demasiada inercia, pero modificando los valores del cuerpo físico del objeto esto se pudo evitar.
También se encontró que al atacar, los proyectiles empujaban al personaje que los lanzaba. Para solucionar esto se puso al proyectil y al personaje en capas de interacción física diferentes, eliminando así su interacción. Puesto que el personaje no se tiene que poder disparar a sí mismo, esto no supone un problema.

Diseñar el arte del juego no ha sido fácil, pero al trabajar con un estilo pixel art[11] se ha podido conseguir un resultado aceptable.

Se han encontrado otras dificultades a lo largo del proyecto pero estas se pueden resumir en que es la primera vez que se utiliza UNITY y se tuvo que aprender desde cero cómo funciona el motor.

7 Conclusiones.

En este proyecto se ha trabajado en muchas de las tareas que componen la creación de un videojuego: se ha planificado el proyecto, diseñado el videojuego, implementado todos los aspectos del juego, y se ha hecho el testing.

[10] Primera versión completa del programa que satisface la mayoría de requisitos.

[11] Estilo que crea diseños al nivel de píxel, y resalta cada pixel de la imagen.

Trabajar en todas las tareas, desde el scripting al dibujo, ha sido una experiencia enriquecedora, que muestra cuánto trabajo hay detrás de un videojuego.
Lo que ha llevado más trabajo es la implementación de los personajes del juego, cada uno con sus comportamientos y atributos que tienen que funcionar con el resto de elementos.
Durante el proyecto se han realizado algunos cambios en la planificación, y no se han cumplido algunas de las predicciones de tiempos de desarrollo. Esto es debido a dificultades encontradas a lo largo del desarrollo, la mayoría de ellas debidas a un desconocimiento de las herramientas utilizadas.
Quizás si hubiera dejado más tiempo para el aprendizaje del uso del motor y para las fases tempranas del proyecto, se hubiera seguido la planificación de manera más fiel.
La documentación utilizada ha sido muy útil a la hora de definir qué tiene que ser este juego, y también se ha utilizado como una guía de los objetivos a cumplir.

La parte del proyecto más difícil de llevar a cabo han sido todas las interacciones entre los diferentes personajes del juego, sin que aparezca algún error durante la partida.

6.3.1 Conocimientos adquiridos.

Durante el desarrollo de este trabajo he aprendido a utilizar el motor UNITY, programar en C# y sobre todo a planificar y llevar a cabo todas las fases de un proyecto, lo cual creo que me será muy útil en el futuro, y a buscar yo mismo las soluciones a los problemas que encuentro.

6.3.2 De cara al futuro

En un futuro se quiere continuar el desarrollo, sobretodo para acabar la IA, y también, para añadir más contenido, mejorar la presentación, y poder transformarlo, finalmente, en un producto comercial.

8. Agradecimientos

Quiero agradecer a mi tutora Katerine Díaz la ayuda prestada durante el proyecto, a mis amigos ynpor ayudarme a testear el juego, y a los usuarios de los foros de UNITY, que siempre están dispuestos a responder las consultas de la gente.

9. Bibliografía

[1] https://www.gamesindustry.biz/articles/2018-01-31-games-industry-generated-usd108-4bn-in-revenues-in-2017 [Consulta: 8-03-2018]
[2] Modelo de desarrollo Scrum https://www.scrum.org/resources/what-is-scrum [Consulta: 8-03-2018]
[3] Documentación en Videojuegos: Documento de diseño (GDD) https://eldocumentalistaudiovisual.com/2015/02/06/documentacion-en-videojuegos-documento-de-diseno-gdd/ [Consulta: 8-03-2018]
[4] Tutorial Unity 2D. https://unity3d.com/es/learn/tutorials/s/2d-game-creation [Consulta: 8-03-2018]
[5]http://sitn.hms.harvard.edu/flash/2017/ai-video-games-toward-intelligent-game/ [Consulta: 1-05-2018]]
[6]https://es.wikipedia.org/wiki/Hack_and_slash [Consulta: 8-03-2018]
[7] Tutorial de Unity para añadir colisiones https://unity3d.com/es/learn/tutorials/projects/2d-ufo-tutorial/adding-collision?playlist=25844 [Consulta: 2-04-2018]
[8] Tutorial de Unity sobre tilemaps. https://blogs.unity3d.com/es/2018/01/25/2d-tilemap-asset-workflow-from-image-to-level/ [Consulta: 1-04-2018]
[9]https://www.aseprite.org/ [Consulta: 2-04-2018]