TOWER SHIFT

Autor: Iñigo Soba Jiménez
Tutor: Helio Tejedor Navarro
Profesor: Joan Arnedo Moreno

Máster Universitario en Diseño y Programación de Videojuegos
Diseño de experiencias de juego

18/06/2023

Créditos/Copyright

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Tower Shift*
Nombre del autor:	*Iñigo Soba Jimenez*
Nombre del colaborador/a docente :	*Helio Tejedor Navarro*
Nombre del PRA:	*Joan Arnedo Moreno*
Fecha de entrega (mm/aaaa):	*06/2023*
Titulación o programa:	*Máster Universitario en Diseño y Programación de Videojuegos*
Área del Trabajo Final:	*Diseño de experiencias de juego*
Idioma del trabajo:	*Español*
Palabras clave	*TFM, videojuego, Unity2D, metroidvania, puzles*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo*

Trabajo de fin de máster que consiste en desarrollar un pequeño videojuego funcional en Unity. Este juego es del género *metroidvania*. Se pretende que este juego ofrezca una jugabilidad bien planteada. También se realiza un estudio de mercado para ver como son el público objetivo y la competencia en este género. Al finalizar se espera tener un pequeño juego 2D completo.

- Enlace al repositorio con el ejecutable y el código fuente: https://gitlab.com/Abos_master/tfm-la-torre

Abstract (in English, 250 words or less):

End of Master Project consisting of developing a small, functional videogame in Unity. This game is of the *metroidvania* genre. The aim is to offer a correctly designed gameplay. A market study is also done to analyse both target audience and competitors. By the time is has been finished, it is expected to have a small and complete 2D game.

- Link to the repository with the source code and executable files: https://gitlab.com/Abos_master/tfm-la-torre

Agradecimientos

A mi familia, por darme la oportunidad de hacer esto.

A mis amigos, por aguantarme estos meses de trabajo.

Y especialmente a Iker, quien me ha ayudado con el tráiler del juego.

Índice

Figuras y tablas

Índice de figuras

1. Introducción

1.1. Introducción/Prefacio

El género *metroidvania* es uno que en los últimos tiempos ha ganado popularidad gracias a la escena *indie*, con ejemplos como Hollow Knight o Blasphemous. Sin embargo, muchos de estos repiten patrones de diseño: habilidades similares, puzles parecidos…

En este proyecto, Tower Shift, se toma en consideración esa cuestión y se busca el crear una experiencia de juego que sea diferente a otras opciones del género. Con la creación de mejoras diferentes a las usuales para así ver que nuevos retos y maneras de jugar se le pueden ofrecer a los jugadores.

El objetivo, pues, es la de crear un juego donde se rompan los patrones establecidos en este tipo de juegos, crear nuevas experiencias para los jugadores, y ver donde puede llevarse el género *metroidvania.*

1.2. Descripción/Definición

En la actualidad existen en el mercado una amplia variedad de juegos *metroidvania*, cada uno con una ambientación y estética diferente.

Aun así, es posible observar patrones en cuestiones como en las mejoras que se obtienen. Cosas como el doble salto o los avances rápidos son muy habituales, asi como el incorporar elementos de juegos como Dark Souls al combate.

Tower Shift intenta alejarse de esa tendencia. La mecánica principal, la inversión de la gravedad, pretende sustituir al clásico doble salto para llegar a lugares más altos. También se ha simplificado el combate, con la intención de volver a un modo de juego más clásico y enfocado en otros aspectos como la exploración.

Con ello este proyecto pretende explorar distintos diseños de niveles que apliquen estos cambios antes mencionados, como una forma de alejarse de las convenciones que han ido surgiendo en el género y para llegar a nuevas cotas.

1.3. Objetivos generales

1.3.1. Objetivos principales

- Crear un juego donde se presenten de manera orgánica sus mecánicas.
- Diseñar un entorno y mecánicas interesantes.
- Presentar un nivel de juego completo.
- Crear un jefe de nivel que desafíe a los jugadores.
- Crear un entorno de juego interesante para el jugador.
- Presentar un sistema de juego atractivo para los jugadores.

1.3.2. Objetivos personales

- Desarrollar un juego *metroidvania*.
- Aprender a manejar el desarrollo de un videojuego desde cero.
- Superar la titulación

1.4. Metodología y proceso de trabajo

El propósito de este trabajo es el de adaptar las ideas de diseño habituales en el género para crear un producto nuevo que sea capaz de diferenciarse del resto.

Debido al tiempo y recursos disponibles, no se pretende crear un producto revolucionario, solo uno que pueda llevar más allá algunas de las convenciones habituales.

Para el desarrollo del proyecto se sigue un desarrollo incremental, donde cada cierto tiempo se entregue una nueva versión del juego con nuevas mecánicas y niveles implementados.

El proyecto se desarrolla en Unity 2D, con scripts programados en C#.

El desarrollo del proyecto está dividido en dos partes principales: En primer lugar, la fase de diseño, donde se han creado los distintos niveles del juego, mecánicas y los enemigos principales, y por otro la implementación y programación de todo.

En la parte de programación se ha optado por un entorno Unity debido a la experiencia obtenida en éste a lo largo de la titulación.

Si bien existen muchos perfiles dentro de un desarrollo, se ha optado por priorizar estas dos partes y conseguir todo lo relacionado con el arte del juego de terceros, ya sea desde la Asset Store de Unity, páginas como Itch.io o videos de plataformas como YouTube.

1.5. Planificación

Figura 1: Planificación del proyecto

1.6. Estructura del resto del documento

En esta sección se va a hablar del contenido de los siguientes capítulos del informe.

- Análisis de mercado: se habla del estado del arte con relación al género y el público objetivo del juego desarrollado.
- Propuesta: objetivos del proyecto a nivel de mercado.

- Diseño: arquitectura de las clases y sistemas del juego, diseño de las interfaces y lenguajes de programación utilizados.
- Implementación: Requisitos de instalación y controles del juego.
- Conclusiones y líneas de futuro: Comentarios finales sobre el trabajo e ideas para continuar con el proyecto.

2. Análisis de mercado

2.1. Público objetivo y perfiles de usuario

En los últimos años, el mercado de los videojuegos ha visto un aumento de la popularidad de juegos que heredan ideas de clásicos como *Castlevania: Simphony of the Night o Super Metroid.* Si bien los grandes estudios no suelen desarrollar títulos de este estilo todavía, este género ha encontrado su lugar entre los desarrolladores indie, donde es notable el aumento de desarrollos a lo largo del tiempo.

El punto fuerte de estos es que se pueden combinar con características de otros géneros con facilidad, lo que hace que pueda haber amplia variedad entre títulos y que puedan interesar a muchos tipos de personas diferentes.

En Tower Shift se hace mucho énfasis en la exploración del entorno, por lo que jugadores que disfruten de juegos donde el movimiento del personaje por los escenarios sea importante.

Por otra parte, como el combate es relativamente simple, no es un juego adecuado para jugadores que busquen un reto a la hora de luchar contra enemigos difíciles.

Por lo tanto, este proyecto va dirigido a los siguientes grupos:

- Jugadores que sigan de cerca la escena indie.
- Jugadores a los que les gusta el género de *metroidvania* o acción y exploración.
- Amantes de los juegos 2D.
- Amantes del estilo *píxel-art* clásico.
- Amantes de los elementos de ciencia ficción.
- Jugadores que no busquen un reto complicado.

2.2. Competencia/Antecedentes

El primer gran exponente del género de exploración en 2D es Metroid, de 1986. Si bien hay algunos juegos anteriores que pueden considerarse también, como Xanadu: Dragon Slayer II, este es el primer gran juego considerado como *metroidvania.*

Figura 2: Metroid (1986)

Este juego es el primero en el que el desplazamiento es libre. En juegos de scroll lateral anteriores, era habitual avanzar a través de los niveles desplazándose a la derecha, en niveles independientes que presentaban desafíos variados. En el caso de Metroid, el mundo de juego está constituido por un único nivel extenso dividido en estancias pequeñas. Para explorar el mundo tenemos la libertad de movernos a izquierda o derecha, y se hace énfasis en aprender a orientarse por el entorno para encontrar nuevos caminos. La exploración libre se puede ver en el mismo inicio del juego, donde si vamos a la derecha pronto nos topamos con un callejón sin salida rápidamente, mientras que por la izquierda está el *power-up* que nos permite avanzar. Esto lleva a que es desarrollo de la aventura fuese no-lineal, una característica que se volvería una de las señas de identidad del género.

Al mismo tiempo que Metroid nacía su franquicia hermana: Castlevania. Esta es la otra que da nombre al género. Concretamente, fue con la salida de Castlevania II: Simon´s Quest de 1987 que la saga adoptó la no linealidad de Metroid.

Figura 3: Castlevania II Simon's Quest

En 1994 saldría Super Metroid, para muchos el pináculo del género. En este juego se veía la fórmula refinada, se añadía inventario y mapa al juego y la libertad a la hora de explorar y abordar los jefes del juego fue la más grande vista hasta el momento.

La mecánica principal está inspirada en Bloodstained: Ritual of the Night, un juego del creador de Castlevania: Simphony of the Night, donde en la recta final del juego se obtiene un poder con el que dar la vuelta a la habitación y recorrerla por el techo. Debido a que esa habilidad aparece al final del juego, en Tower Shift se pretende diseñar todo el juego alrededor de este poder.

Figura 4: Bloodstained

Otro de los juegos que más han influido en Tower Shift es Hollow Knight, que marcó un nuevo estándar para los videojuegos independientes y que también marcó la nueva tendencia que han seguido juegos del mismo género después de él.

Más juegos que han inspirado este proyecto incluyen Axiom Verge, un juego muy inspirado en los Metroid clásicos, y Blasphemous, que le da más importancia al combate para ofrecer un título desafiante.

2.2.1. Inspiración actual

En la actualidad los metroidvania están en auge y hay muchas opciones en el mercado. Por lo tanto, podemos obtener inspiración para el proyecto de estos juegos. Estos son algunos de esos juegos.

HOLLOW KNIGHT (2015)

Posiblemente uno de los juegos independientes más reconocido actualmente. Hollow Knight pone mucho en énfasis en la exploración, no solo para obtener secretos, sino que el propio desarrollo de la historia del juego no es lineal y cambiará dependiendo de que hagamos en nuestra partida.

La exploración que ofrece este juego es impresionante, y Tower Shift pretende imitar ese aspecto del juego creando un diseño de niveles donde el jugador pueda llegar al objetivo por varios caminos, que tenga zonas opcionales a las que llegar con ingenio y obtener recompensas por ello…

Figura 5: Ilustración de Hollow Knight

AXIOM VERGE (2015)

Este juego, desarrollado por una única persona, es un *metroidvania* que se inspira totalmente en Metroid. Se puede ver que tanto el apartado gráfico como las mecánicas son muy similares a la longeva saga de Nintendo.

El escenario de ciencia ficción de Axiom Verge inspira en gran medida el escenario de Tower Shift. También es la principal inspiración de los gráficos del juego, su estilo con *tiles pixel-art* que busca emular a los Metroid clásicos.

La ambientación del juego es similar a lo que se busca en este proyecto: una sensación de misterio, donde la historia va llegando poco a poco y el jugador se siente solo y aislado en un entorno que no es familiar para él.

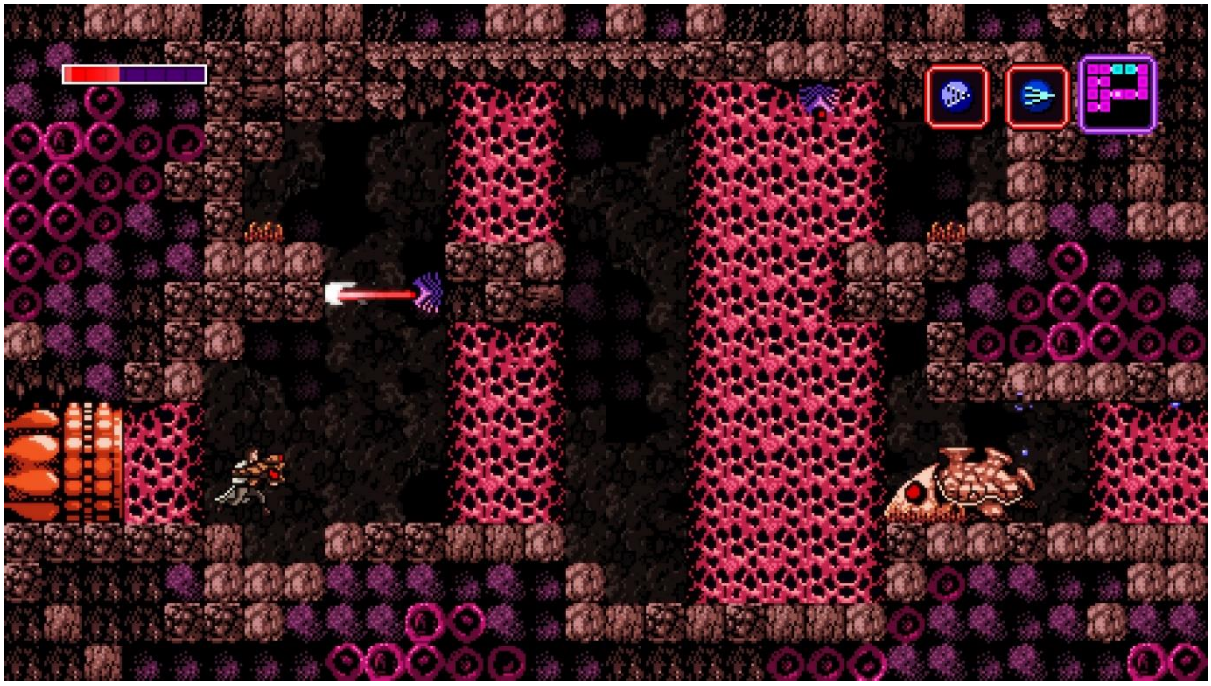

Figura 6: Pantalla de juego de Axiom Verge

BLASPHEMOUS (2019)

Del estudio español The Game Kitchen, este juego toma inspiración en la cultura del sur de España. Blasphemous ofrece una desafiante mezcla de plataformeo y combate, donde esto último recuerda a juegos como Dark Souls. Aun asi, no se deja de lado la exploración; el mapa es complejo y está interconectado, además de darnos la posibilidad de encontrar muchos secretos por él.

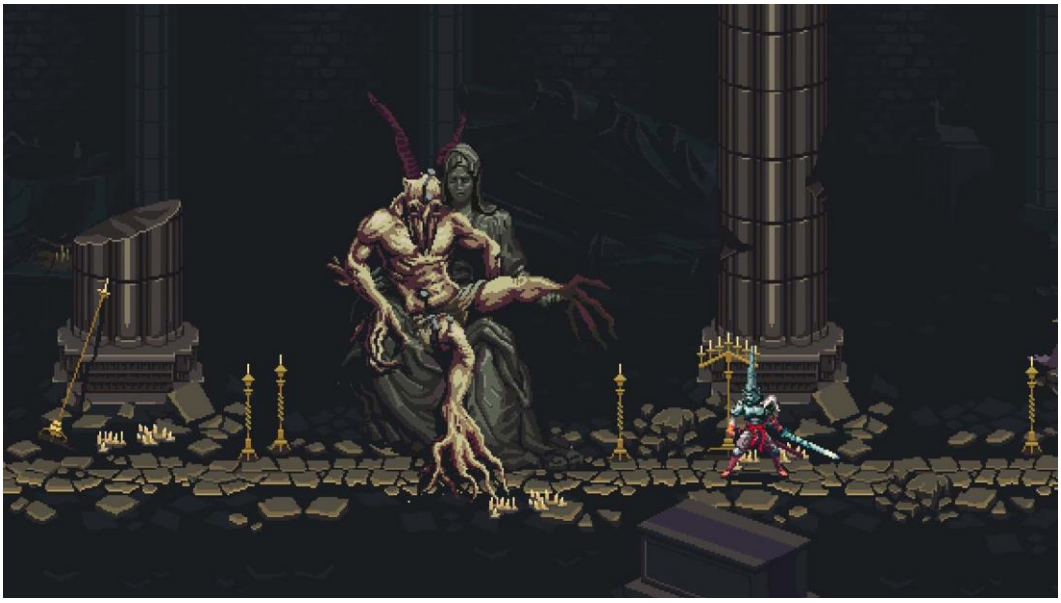

Figura 7: Pantalla de juego de Blasphemous

2.3. Análisis DAFO

- Mecánicas novedosas que interesen a los jugadores. - El reto puede llamar la atención de ciertos jugadores.	- El aumento de la popularidad de este tipo de juegos juega a su favor. - La mecánica principal no es habitual y puede atraer jugadores.
- Género aún bastante nicho en el mercado. - La dificultad puede no atraer al grupo más casual de jugadores.	- Hay muchas opciones del género y es difícil destacar. - Muchos de los juegos son sobresalientes por lo que hace falta un producto de mucha calidad.

3. Propuesta

Como ya se ha mencionado anteriormente, Tower Shift es un juego de género metroidvania (acción y exploración 2D). El objetivo del juego consiste en llevar al personaje principal a través del escenario del juego, en este caso una torre, superando las habitaciones

que la componen. En estas estancias se pueden encontrar multitud de cosas: enemigos que derrotar, nuevas habilidades, retos que superar usando dichas habilidades…

3.1. Definición de objetivos/especificaciones del producto

En esta fase del producto se desarrolla el primer tramo o nivel del juego, que abarca desde el comienzo hasta enfrentar al primer jefe. Durante este recorrido se obtiene la habilidad principal del juego y se prueban las capacidades que está otorga.

3.2. Personajes del juego

Durante el desarrollo de la demo los jugadores se encontrarán con algunos personajes. Aunque en la demo actual no se ve ningún elemento narrativo principal, se puede hablar del protagonista y de los enemigos que encontrarán los jugadores.

3.2.1. Personaje jugable

El personaje principal del juego es una persona armada con una espada que despierta dentro de la Torre sin saber cómo ha llegado hasta ahí. Valiéndose de su arma y las habilidades que irá obteniendo por la Torre deberá intentar encontrar una salida para poder escapar.

3.2.2. Personajes enemigos

Los enemigos que aparecen en la primera versión del juego son todos mecánicos. En primer lugar, un enemigo bomba cuyo único ataque es explotar delante del jugador. Otro que tenemos es el murciélago, que vuela recorriendo la habitación e intenta chocarse con el jugador. También hay un robot araña que se acercará al protagonista y le atacará de frente. Este último es más fuerte que los anteriores.

Por último, el jefe del primer nivel del juego es un Golem con varios ataques en su arsenal. Primero lanza su brazo hacia donde esté el jugador. Después lanza un rayo láser desde su ojo que recorre el suelo y obliga al jugador a esquivarlo saltando. Al final, el MechaGolem puede entrar en un breve periodo donde se protege y no se le puede hacer daño.

Cuando se le hace daño suficiente, el jefe pasa a la segunda fase. Aquí el ataque láser sube por la pared, lo que obliga a ser más preciso todavía a la hora de esquivarlo.

3.3. Mecánicas

Las mecánicas son la parte crucial de un videojuego, y la demo de Tower Shift presenta algunas de ellas:

3.3.1. Mecánica principal: Inversión de gravedad

La mecánica principal de Tower Shift es la de invertir la gravedad. Es habitual en juegos de este género tener habilidades como doble salto o un *dash* para alcanzar zonas inaccesibles. En este caso, se pretende ofrecer una mecánica diferente que haga que los jugadores piensen de manera diferente a lo que están acostumbrados en los *metroidvania*.

Por otra parte, este juego cuenta con combate, pero como se centra en la exploración éste es básico y apenas tiene profundidad.

3.3.2. Exploración

En cuanto a la exploración, este título hace uso extenso del llamado *backtracking*, término que describe cuando en un juego se debe volver atrás sobre nuestros pasos para avanzar u obtener algún secreto. El *backtracking* es muy habitual en este género, usado para poner énfasis en la exploración del mapa disponible y para hacer que el desarrollo de la aventura sea lo menos lineal posible. En el caso de Tower Shift, los jugadores podrán encontrar zonas a las que tendrán que volver más adelante para avanzar o mejorar al personaje, mediante el uso de nuevas habilidades que vayan obteniendo o de atajos que lleguen a abrir.

3.4. Escenario del juego

En un principio, el entorno del juego hace uso de una estética de ciencia ficción, con entornos metálicos y enemigos robóticos. Conforme el jugador va avanzando por el escenario, la temática va incorporando elementos más naturales (paredes de roca, vegetación…) para acabar en una mezcla de lo natural y artificial que muestre lo extraña que es la Torre.

El jefe del primer nivel es el mayor ejemplo de esto, pues a pesar de ser claramente tecnológico está compuesto de roca en su mayoría.

3.5. Estrategia de marketing

Con el objetivo de dar a conocer el producto al mayor número de personas posibles, una sólida estrategia a seguir es aprovechar las redes para conseguir la mayor difusión posible.

Lo primero es elegir donde se publicará el juego. Existen varias plataformas donde hacerlo, siendo la más conocida Steam. Aun asi, debido a que publicar un juego en Steam requiere una inversión inicial, no sería la opción más adecuada para un nuevo desarrollador.

En su lugar, se busca publicar Tower Shift en Itch.io. Está página permite publicar juegos sin ninguna inversión inicial, lo que elimina el problema de Steam. Además, también permite el que sean gratuitos. Gracias a esto se puede empezar con una demo gratuita del juego para que la gente pueda probarlo, después aprovechar el alcance obtenido al ponerlo a la venta más adelante, con el añadido de que esa demo serviría para encontrar cosas que mejorar antes de la salida oficial del producto.

Una vez obtenida una repercusión considerable con el método anterior se puede pasar a publicar el juego en otras plataformas como Steam o Epic Store, donde llegarán al grueso de jugadores de PC.

Otra estrategia que seguir es la de entregar claves anticipadas del juego a *youtubers* y *streamers* para que estos lo jueguen en directo y lo muestren a su público.

Por último, está la creación de una cuenta oficial para redes sociales en la que compartir información a lo largo del desarrollo, con el objetivo de mantener un contacto con la comunidad y generar expectación.

4. Diseño

4.1. Arquitectura general de la aplicación/sistema/servicio

4.2. Diseño del nivel

El juego comienza en un pasillo por el que solo se podrá avanzar de frente. Esta habitación tiene algunos desniveles que permiten probar el salto del personaje. Una vez llegado a la primera puerta, la siguiente estancia es una torre con plataformas por las que se puede ir subiendo mediante el salto. En esta zona ahí varios caminos por los que ir, aunque con las habilidades que se tienen al principio no se puede avanzar por la mayoría de ellos. En la zona inferior se encuentra una sala donde es posible guardar la partida.

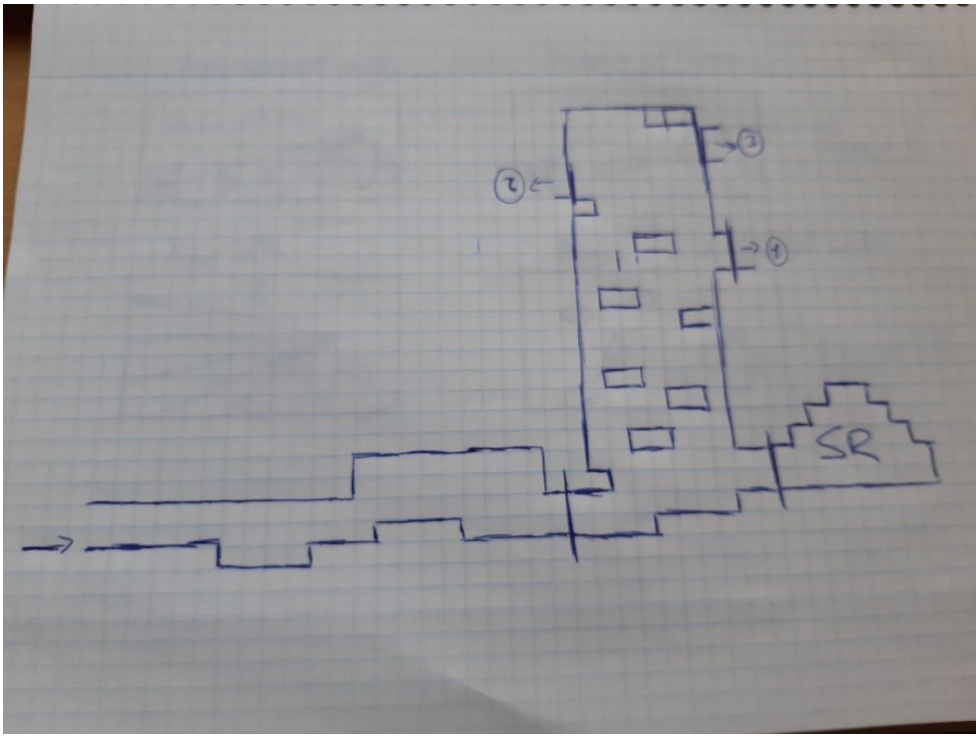

Figura 8: Primeras salas

La siguiente habitación por la que se pasa hacia la izquierda contiene una zona elevada a la que no se puede acceder debido a que el salto no llega lo bastante alto, por lo que se continúa avanzado de frente hasta la puerta.

Se sigue hasta llegar a una habitación sin salida donde se encuentra el primer “powerup” del juego: el inversor de gravedad. Con esto el jugador puede ahora invertir el efecto de la gravedad en él para subirse a los techos de las estancias.

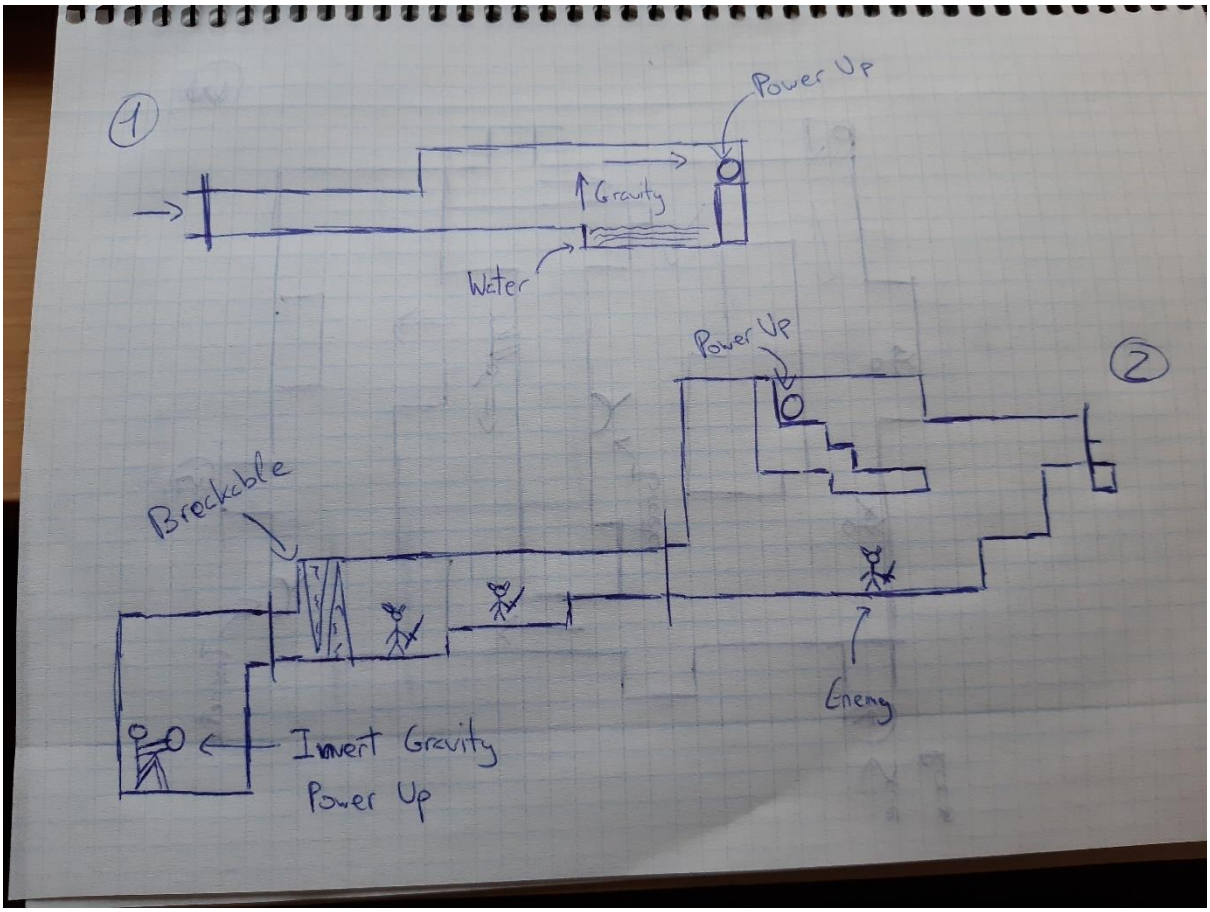

Figura 9: Zona del inversor de gravedad

Con esta habilidad se puede volver a las zonas anteriores y recoger algunas mejoras para el personaje, como en la zona a la que no se podía llegar saltando en la habitación anterior.

De vuelta en la sala parecida a una torre, gracias a la nueva habilidad ahora se puede alcanzar la puerta de la zona superior y seguir el camino.

En esta nueva habitación habrá primero que dejarse caer por el hueco. Ya abajo habrá dos caminos disponibles: por el de arriba se encuentra una torreta en el techo que disparará cuando el jugador se acerque, y que se puede destruir atacándola desde el techo. Por el de abajo el jugador deberá enfrentarse a dos enemigos que atacarán de cerca.

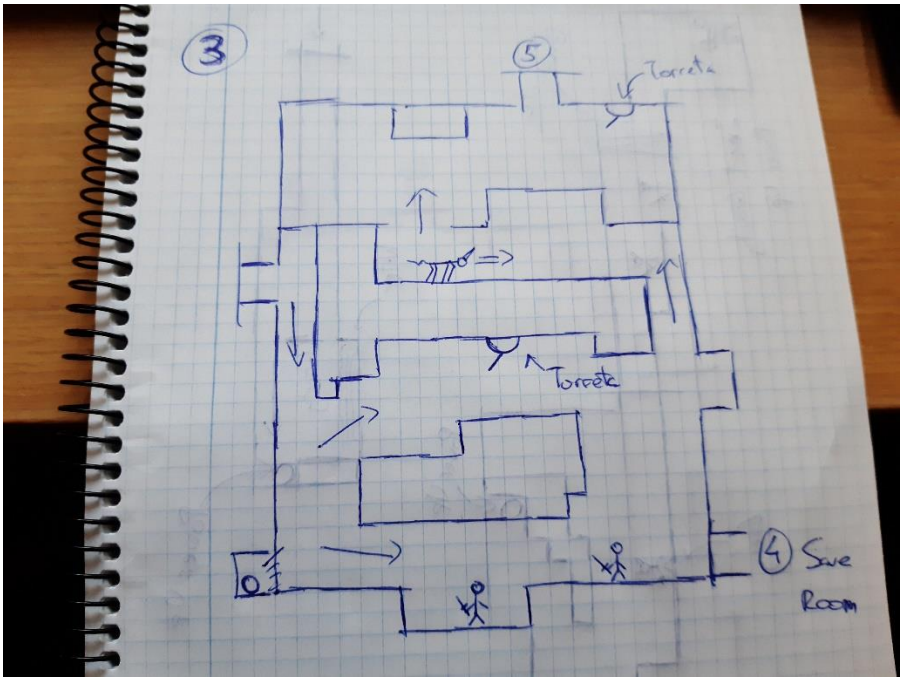

Figura 10: Sala de las torretas

En la zona inferior izquierda se puede encontrar una mejora para el personaje, tan solo hay que atravesar la pared falsa que estará justo enfrente.

Una vez pasado este primer tramo, lo siguiente será utilizar la inversión para colarse por el hueco del techo y avanzar por el pasillo siguiente, donde está colocada otra torreta como la anterior. Al final de todo se debe pasar por una puerta situada en la zona superior, a la que solo se puede acceder mediante la inversión.

En esta nueva habitación se podrá avanzar a ambos lados; por la izquierda se vuelve a la torre, el camino principal va por la derecha. Se puede ver que se debe subir mediante la inversión hacia el techo. Pero esta vez en el camino del jugador hay colocados unos láseres que le obligarán a calcular correctamente cuando pasar. Una vez superado este obstáculo se pasa a la siguiente estancia.

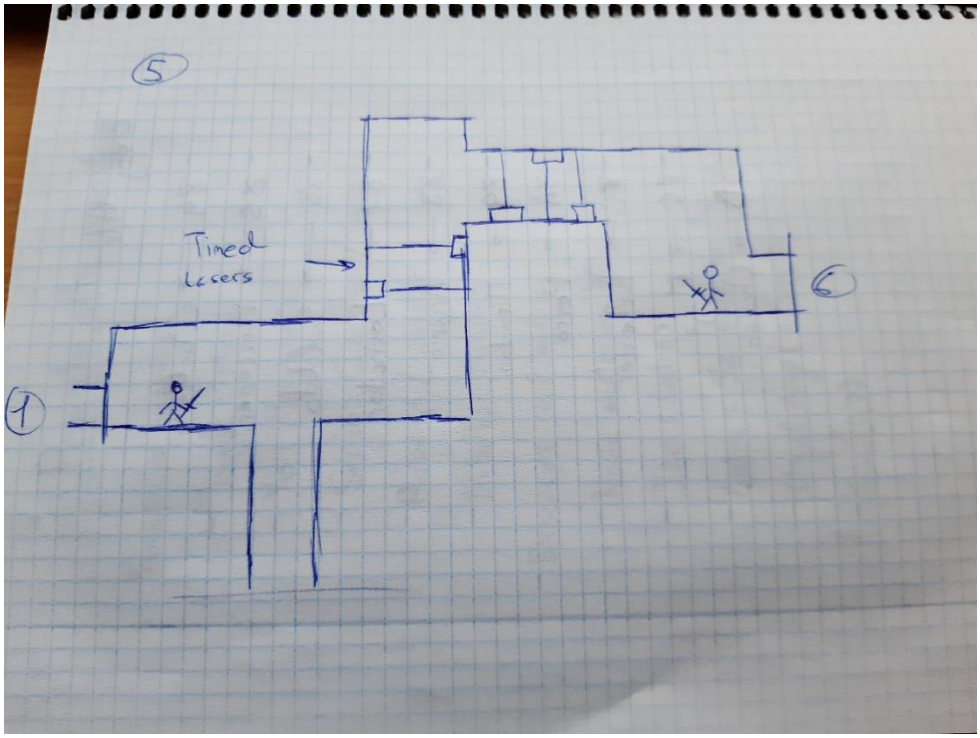

Figura 11: Sala de los láseres

Si se toma el camino de la izquierda para volver a la sala que parece una torre, podremos llegar hasta un dispositivo que permite abrir un atajo. Con este nuevo camino abierto ya no se debe dar un gran rodeo para llegar hasta aquí desde el inicio. También se puede utilizar para volver atrás y obtener mejoras que se hayan os dejado por el camino.

A la izquierda de esta sección hay otra habitación. Aquí se debe superar un pequeño reto que consiste en superar las plataformas móviles para obtener una mejora para el personaje.

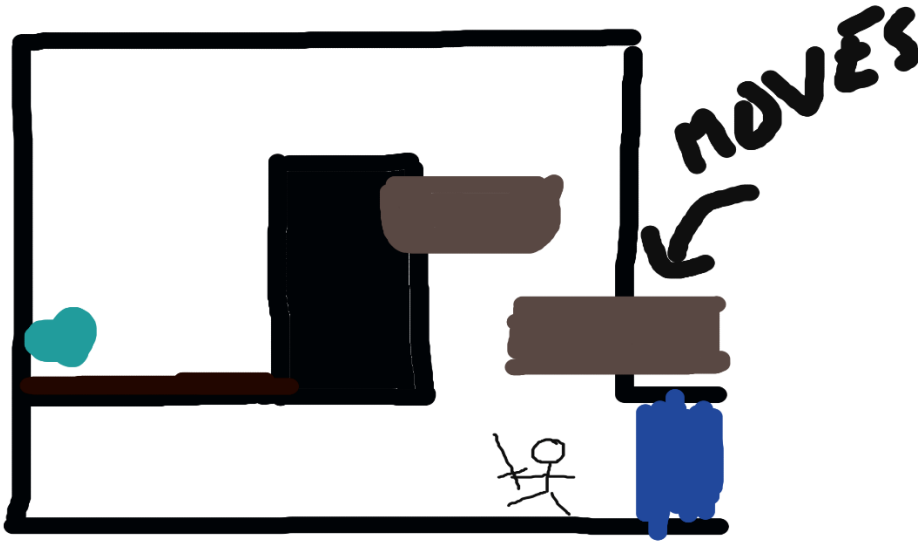

Figura 12: Sala de plataformas móviles

La siguiente habitación trata de superar un foso por el que no se puede cruzar normalmente. Para ello hay que valerse de una plataforma que lo recorre de lado a lado. Durante el recorrido hay obstáculos que hay que evitar saltándolos y evitando perder la plataforma. Además, en el techo se encuentras plantas que dañan al personaje al tocarlas para que no podamos evitar todo pegados al techo.

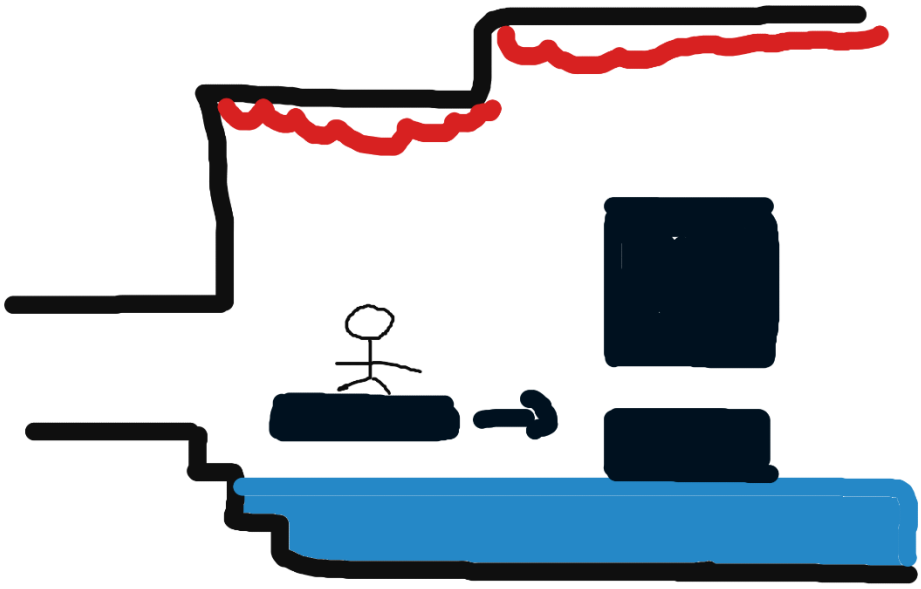

Figura 13: Primera parte de la habitación de la plataforma

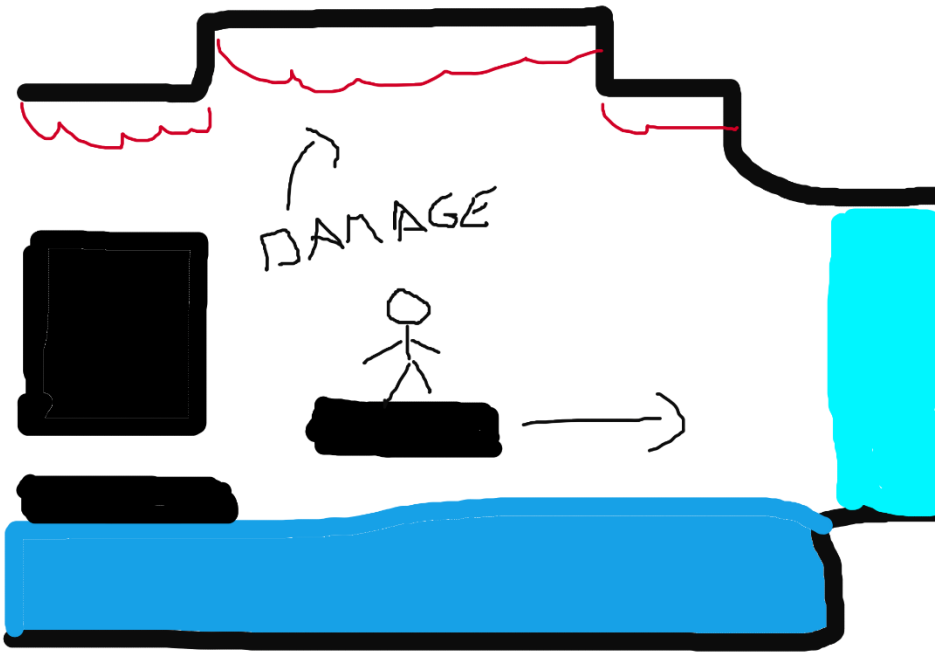

Figura 14: Segunda parte de la habitación de la plataforma

En la última habitación se encuentra el primer jefe del juego: el MechaGolem. Este jefe ataca de dos maneras diferentes. Una es disparar su brazo hacia el jugador, que se puede esquivar fácilmente evitando su trayectoria. El segundo consiste en un láser que dispara desde su ojo y que recorre el suelo de la sala. Este se puede evitar saltando o usando la inversión si no se quiere arriesgar. Cada cierto tiempo el gólem se protegerá, lo que hará que evite todo el daño por unos segundos.

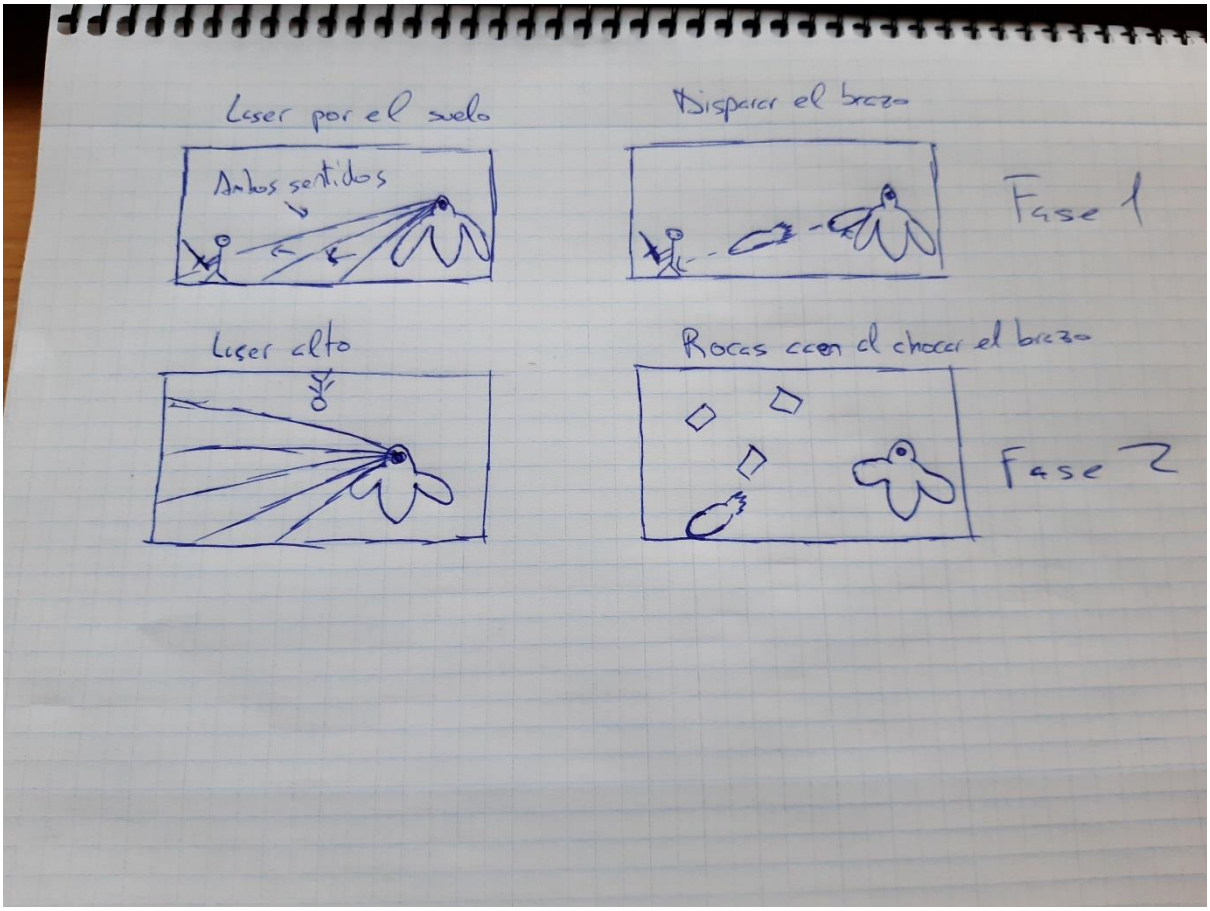

Figura 15: Estados del MechaGolem

4.3. Diagramas de clase

El proyecto está compuesto por una gran cantidad de clases que interactúan entre ellas. A continuación, se verán los diagramas de estas clases.

4.3.1. Personaje Jugable

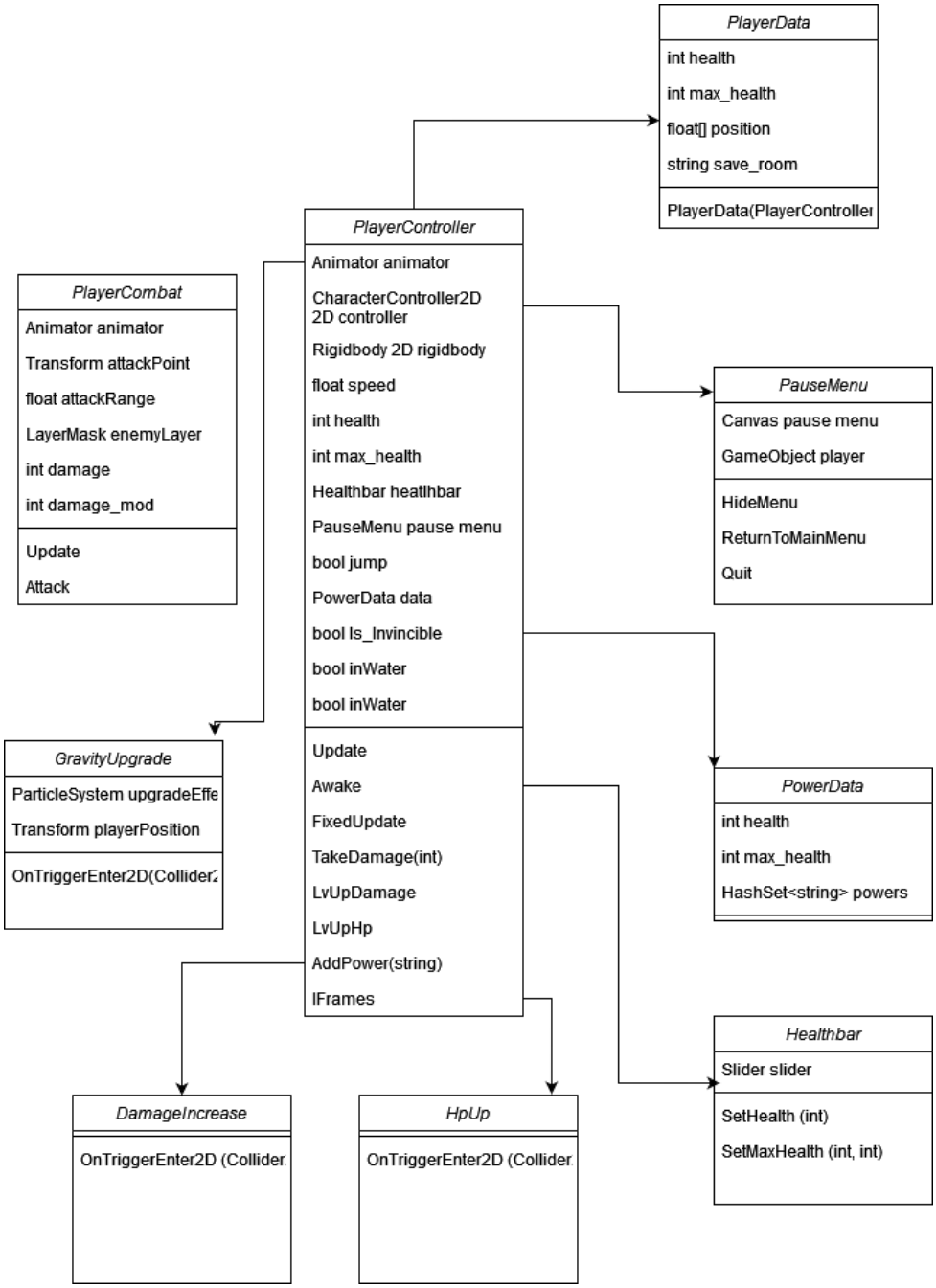

Figura 16: Clases del jugador

PlayerController es la clase más importante, la que define como actúa el personaje jugable. En esta se definen variables como la velocidad de movimiento, la vida total y actual del jugador, su conjunto de poderes…

En cada frame se comprueban los inputs que se reciben para realizar la acción correspondiente, ya sea movernos, saltar o usar un poder. Después en FixedUpdate usamos el método Move del controlador para aplicar el movimiento.

El método TakeDamage es llamado por los objetos que colisionan con el jugador, como los enemigos y los proyectiles para realizar el daño. En relación con esto la corrutina IFrames se usa para que el jugador tenga un par de segundos de invulnerabilidad entre daños y pueda reaccionar.

4.3.2. Enemigos

Bomb

El enemigo más simple del juego, se limita a esperar a que el jugador entre en su rango, y una vez que lo haga, lo persigue por el suelo. Cuando toca al jugador la bomba explota y hace algo de daño.

Su comportamiento está definido por una simple máquina de estados con la que cambia entre esperar o atacar al jugador.

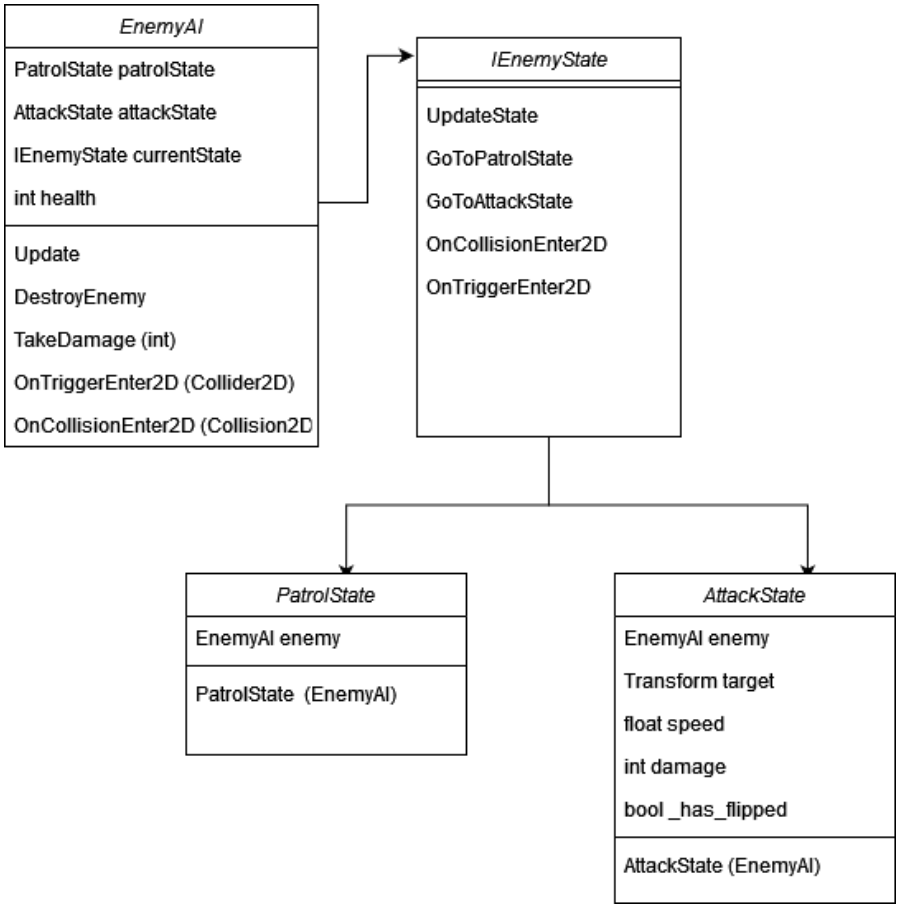

Figura 17: Máquina de estados de la bomba

Bat

Otro enemigo relativamente simple, vuelan en una sola dirección intentando chocar con el jugador. El vuelo sigue un movimiento de onda. Estos aparecen cuando el jugador pasa por ciertas zonas y no lo perseguirán.

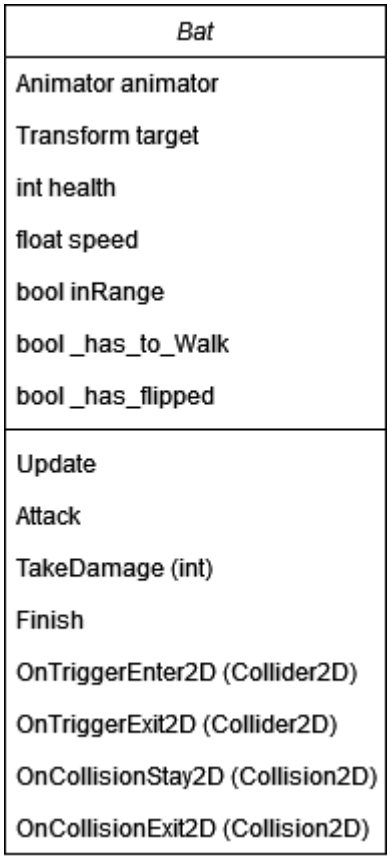

Figura 18: Clase Bat

Spider

Las arañas son más duras que los anteriores, se acercan cuando se entra en su rango y atacan cuando tocan al jugador. Si se deja de tocar la araña persigue al jugador siempre que esté dentro de su rango.

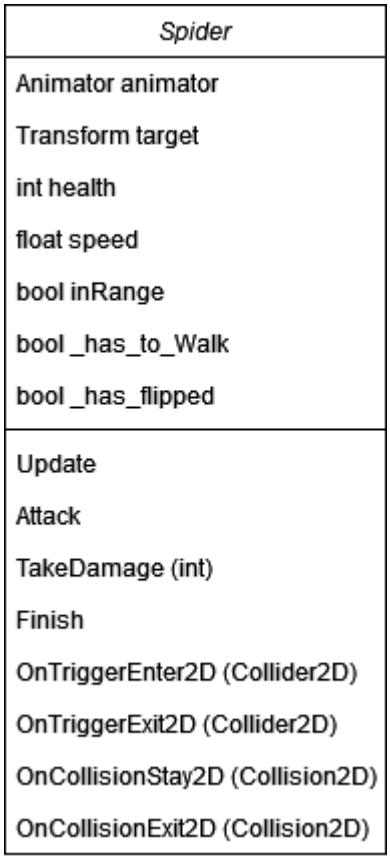

Figura 19: Clase Spider

Turret

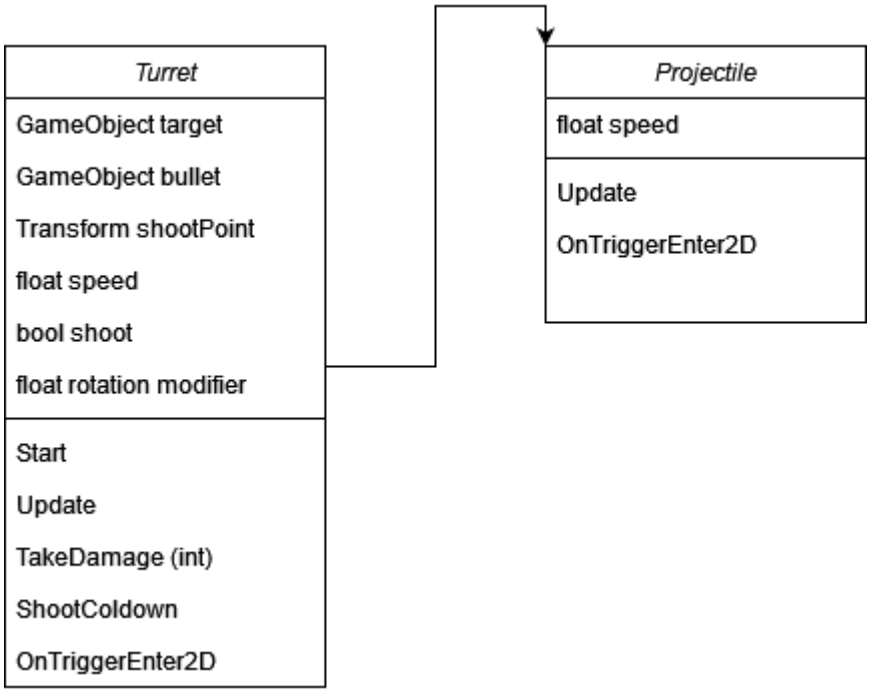

Figura 20: Clases del enemigo torreta

La clase Turret es una simple clase que define el comportamiento de las torretas. Estas giran continuamente hacia el jugador siempre que este en su rango de disparo y cada pocos segundos disparan un proyectil. Esta orientación hacia el jugador está controlada de manera que no pase de cierto punto y la torreta se dé la vuelta completa.

Projectile simplemente avanza en línea hasta que colisiona con algo, y en caso de ser el jugador lo daña.

4.3.3. MechaGolem

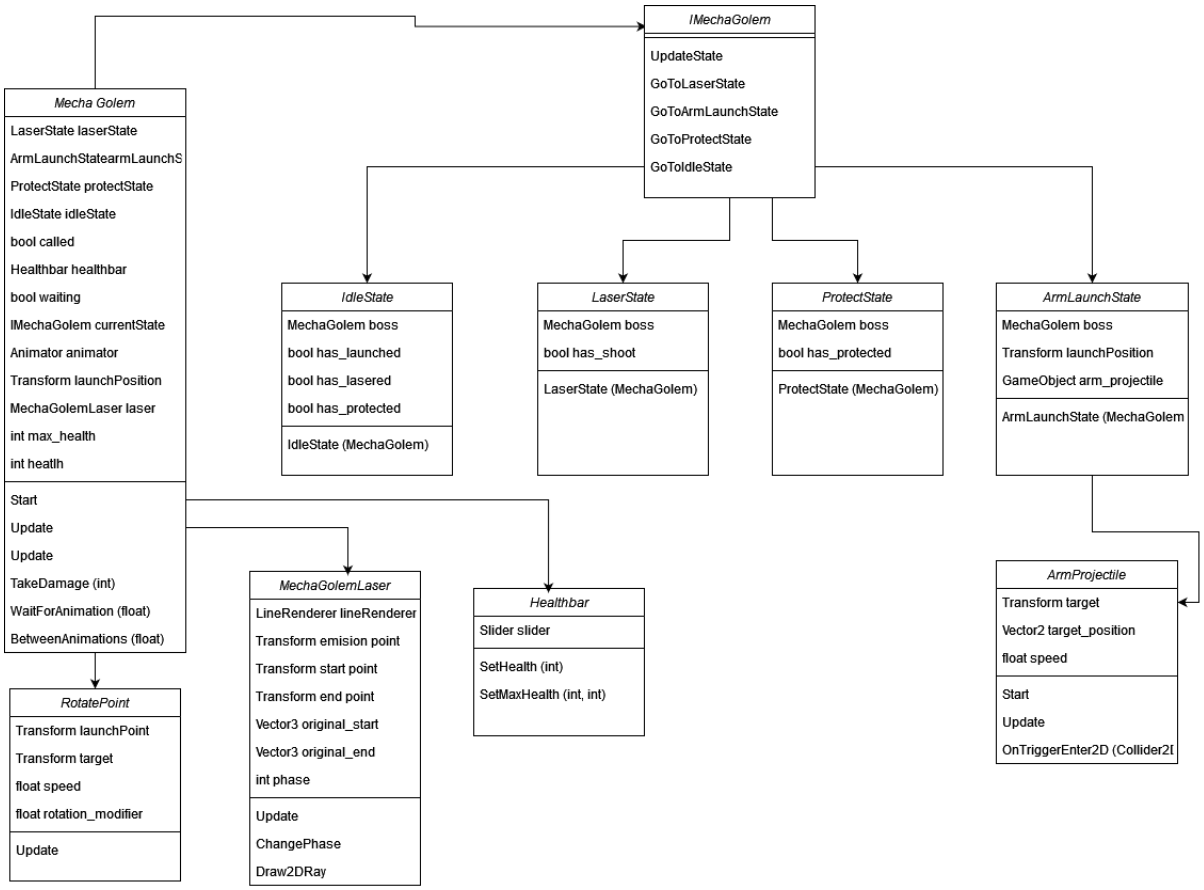

Figura 21: Diagrama de MechaGolem

El jefe del nivel hace uso de una máquina de estados para modelar su comportamiento, indicado por la interfaz IMechaGolem. Este comportamiento se divide en 4 estados.

Idle actúa como un estado intermedio entre los demás donde no se realizan ataques. Pasados unos segundo pasa a uno de los estados de ataque: LaserState, ArmLaunchState o ProtectState. Cada vez que uno de estos estados termina vuelve a Idle, y cuando ha pasado por los tres estados vuelve a empezar.

Después están las clases que controlan otros elementos del jefe. Healthbar se encarga de actualizar la barra de vida del jefe que el jugador puede ver en todo momento en base al campo health de la clase principal. MechaGolemLaser define el comportamiento del ataque láser y se encarga de activar o desactivar el componente encargado de ello según la situación. Por último, RotatePoint orienta el punto desde el que sale el brazo disparado de manera que siempre nos apunte a nosotros.

4.3.4. Menú y Entorno

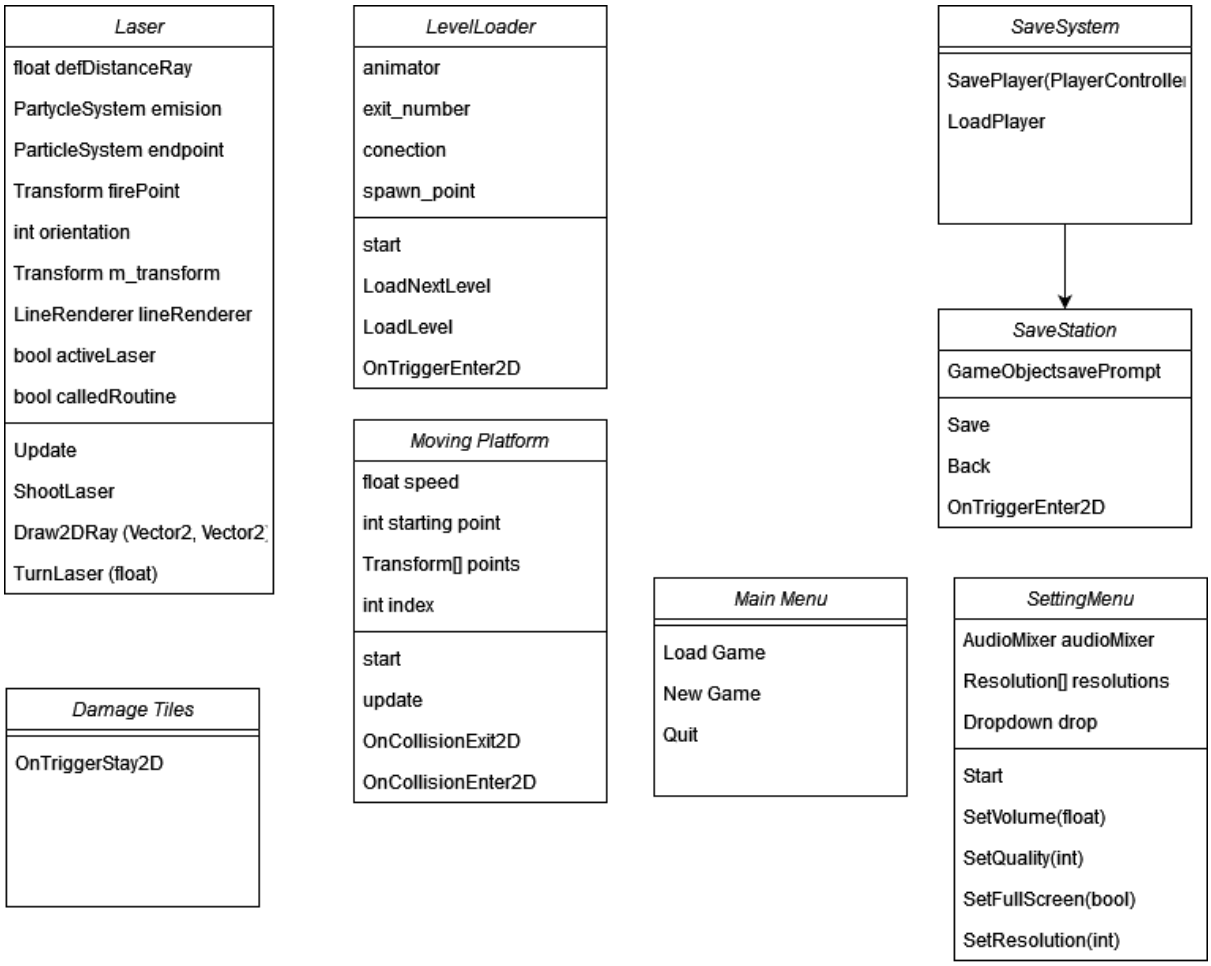

Figura 22: Menús y entorno del juego

Estas son varias clases que se encargan de controlar los menús del juego y los elementos del entorno.

La clase *MainMenu* gestiona el menú principal del juego, desde el que podemos iniciar una partida o cargar una ya existente. Por otro lado, *SettingsMenu* maneja el menú de ajustes que controla el volumen, la resolución…

Laser y *DamageTiles* definen varios elementos del entorno. El primero define los láseres que encontramos y que bloquean el camino cada cierto tiempo. El segundo hace que las zonas que toquemos nos vayan dañando cada poco tiempo.

Las *MovingPlatform* son aquellas que se mueven como si estuvieran sobre raíles y que podemos montar para desplazarnos.

Por último, *SaveSystem* y *SaveStation* controlan el cómo se guarda la partida. El sistema se encarga de la información que se almacena, mientras que las estaciones son las salas del juego donde se puede llevar a cabo esa acción.

4.3.5. Level Conection

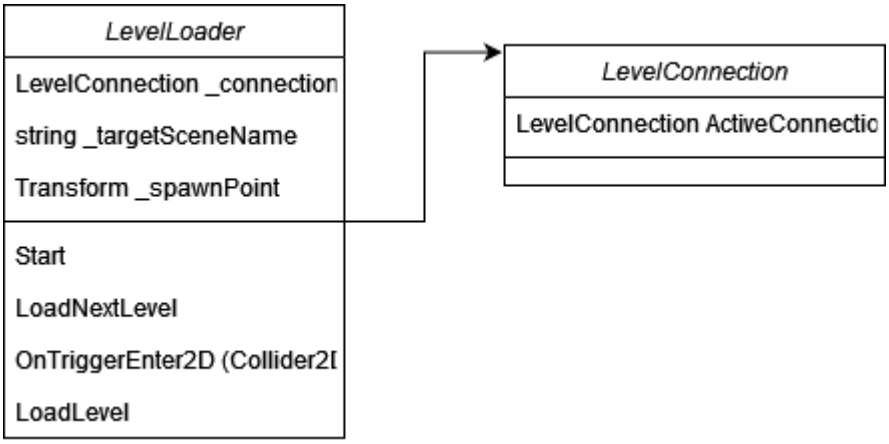

Figura 23: Conexión entre salas

Las conexiones entre salas se utilizan para que el juego sepa, al cargar una nueva habitación, de donde venimos y donde debe colocar al jugador.

Para ello se emplea el *ScriptableObject LevelConnection* para definir una conexión entre dos salas contiguas. Cuando carguemos una estancia, se comprueban todas las conexiones de esa habitación hasta encontrar la que coincida con la que hemos utilizado. Cuando la tenga se coloca al jugador en el punto asignado en el *LevelLoader* correspondiente.

4.4. Lenguajes de programación y APIs utilizados

El desarrollo de videojuegos es multidisciplinar. Eso quiere decir que hace falta usar muchos tipos de herramientas para crear todas las partes que lo componen.

4.4.1. Entorno

Figura 24: Logo de Unity

El entorno utilizado para desarrollar Tower Shift ha sido Unity. Unity es un entorno de diseño de videojuegos que ofrece la posibilidad de realizar proyectos tanto en 2D como 3D. Se ha escogido este motor sobre otras opciones como Unreal por varias razones.

La primera es que es el motor con el que se ha trabajado durante todo el curso, y por tanto en el que más experiencia se tiene. Al contar con tiempo limitado se optó por usar un motor conocido en lugar de tener que aprender las particularidades de alguno nuevo como Unreal Engine.

Otra de las razones es la versatilidad que ofrece Unity respecto a las plataformas para las que se puede desarrollar. Este motor permite desarrollar para PC, las consolas del momento o móvil. Incluso se puede desarrollar para dispositivos VR.

Por último, aunque otros motores pueden llegar a ofrecer más potencia, Tower Shift no requiere de mucha potencia por lo que Unity es más que suficiente para un proyecto en 2D.

4.4.2. Lenguaje de programación

Una vez elegido el motor Unity, hay dos lenguajes que se pueden usar: C# y JavaScript. En este caso se ha elegido usar C# porque, al igual que con el motor, se tiene más experiencia con él, y además porque es más utilizado que JavaScript y se tiene más información en Internet en caso de tener que hacer una consulta.

Como entorno de programación se ha escogido Visual Studio, que funciona en conjunto con Unity.

Figura 25: Logo de Visual Studio

4.4.3. Efectos de sonido

Para los efectos de sonido se ha utilizado un generador online llamado JSFXR. Este es un generador de sonidos de 8 y 16 bits con muchas opciones para generar distintos efectos. También ofrece una opción para generar sonidos aleatorios y después modificarlos como se desee.

Ya que Tower Shift busca imitar los juegos *metroidvania* más clásicos, este simple editor es perfecto para generar los efectos de sonido que se necesitaban.

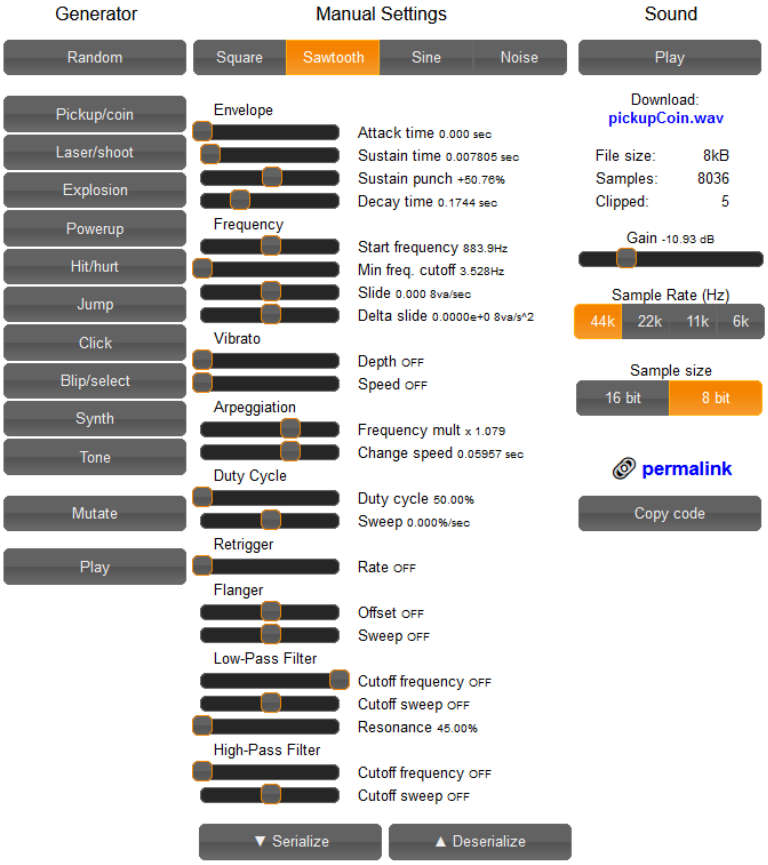

Figura 26: Interfaz de jsfxr

4.4.4. Recursos de terceros

Tower Shift usa muchos recursos de terceros, desde animaciones hasta scripts. A continuación, se presenta la lista de recursos utilizados:

- Tileset: https://s4m-ur4i.itch.io/minimal-futureset-pixelart-2d
- Sprites del personaje jugable: https://heirey.itch.io/district-tileset
- Sprites y animaciones del jefe MechaGolem: https://darkpixel-kronovi.itch.io/mecha-golem-free
- Murcielago: https://pixfinity.itch.io/the-dungeon-pack-1
- Controlador 2D del personaje: https://www.youtube.com/watch?v=dwcT-Dch0bA
- Background generator: https://deep-fold.itch.io/space-background-generator
- sinusoidual move scripts: https://www.youtube.com/watch?v=rHh43ilfnyI
- Sprites y animaciones del enemigo araña: https://vivicat.itch.io/spider-drone
- Script de cambio de escena de LvConnection: https://www.youtube.com/watch?v=r37YoRWYjX0

- Script de plataforma móvil: https://www.youtube.com/watch?v=GtX1p4cwYOc
- Combate a melee: https://www.youtube.com/watch?v=sPiVz1k-fEs&t=190s

5. Implementación

5.1. Requisitos de instalación

Este producto no requiere de nada especial para instalarse en el PC, se puede jugar simplemente teniendo teclado y ratón.

5.2. Controles

- Botones AWSD / Flechas de dirección: Movimiento del jugador
- Espacio: Salto
- Click derecho de ratón: Atacar
- Botón Z: Invertir la gravedad
- Botón M: Menú de pausa
- Botón Enter: Activar mecanismos

6. Conclusiones y líneas de futuro

6.1. Conclusiones

En líneas generales, considero que se han cumplido los objetivos planteados al inicio del proyecto. Durante el desarrollo del producto algo que he aprendido es que la planificación inicial es muy importante. Este ha sido un desarrollo pequeño por lo que no se ha notado mucho, pero es fácil de entender una vez se ha llevado a cabo un proyecto de este tipo que no tener un plan adecuado acarrea grandes problemas a la larga.

Aun se quedan cosas en el tintero. El apartado artístico podría estar más pulido, con mejores animaciones y mayor feedback visual. También falta trabajar más el apartado sonoro.

En cuanto al diseño de niveles y la programación del juego, se han completado los objetivos planteados. Se ha podido programar todas las ideas que se iban desarrollando, y el juego tiene un número de salas jugables considerable.

En general se ha seguido la planificación. No se ha tenido que cambiar nada para lograr la finalización del proyecto. El juego, como ya se ha mencionado, es un proyecto

pequeño por lo que eso ha ayudado a que los cambios y mejoras introducidos durante el desarrollo se han podido realizar de manera flexible y sin muchos problemas.

6.2. Líneas de futuro

Como ya se ha mencionado en la sección anterior, en el futuro se pretende expandir todo el apartado artístico del juego. Con un equipo pequeño se podría llevar a cabo; un par de personas que se encarguen de las animaciones y la música y efectos.

En cuanto a ampliar el contenido, se debería seguir la línea de diseño de las habitaciones de la demo. Estas deberían incluir nuevos puzles que se resuelvan cambiando la gravedad o con nuevas habilidades que se vayan consiguiendo. Relacionado con esto último se podría preparar una pantalla de inventario donde poder ver que habilidades tenemos disponibles.

Para crear esos nuevos escenarios de juego, se seguiría usando la mecánica de la inversión para ello. Por ejemplo, una idea que no llegó a la versión final es la de una sala en la que unos bloques vengan hacia el jugador desde el lado de la sala en dos alturas y el jugador deba cambiar la gravedad para esquivarlos pasando de uno a otro.

Una habilidad que se planteó y que finalmente no está incluida en esta versión es la de enviar una onda de energía por el suelo. Esto tendría dos propósitos: daría otra posibilidad al combate y también serviría para poder accionar mecanismos a los que no pudiéramos llegar. Y después se pueden combinar esto con el uso de la gravedad para otras salas.

Con el tema de la gravedad, ya que bajo el agua no se puede invertir se podría colocar agua en el techo para controlar donde puede lanzarse el jugador. Serviría igual que los elementos dañinos de escenario.

Para ampliar el repertorio de enemigos, se podrían introducir enemigos que puedan perseguir al jugador por el techo, para que invertir la gravedad no permita esquivar a todos los enemigos fácilmente. También se deberían incluir enemigos que puedan atacar a distancia.

En general estoy contento con el resultado obtenido de este trabajo. Es una buena base sobre la que trabajar y dada la oportunidad me gustaría completar el desarrollo y sacarlo adelante como un producto completo.

Bibliografía

Unity – Unity Documentation: https://docs.unity3d.com/Manual/index.html, consultado 15/03/2023 – 08/06/2023

YouTube – 2D Camera in Unity (Cinemachine Tutorial): https://www.youtube.com/watch?v=2jTY11Am0Ig&t=271s, consultado 13/04/2023

YouTube – Create a 2D laser in 60 seconds Unity: https://www.youtube.com/watch?v=vdci2oxVaoA, consultado 20/05/2023

YouTube – How to make AWESOME Scene Transitions in Unity!: https://www.youtube.com/watch?v=CE9VOZivb3I&t=2s, consultado 11/04/2023

YouTube – MELEE COMBAT in Unity: https://www.youtube.com/watch?v=sPiVz1k-fEs&t=190s, consultado 08/05/2023

YouTube – PAUSE MENU in Unity: https://www.youtube.com/watch?v=JivuXdrIHK0&t=15s, consultado 26/05/2023

YouTube – SAVE & LOAD SYSTEM in Unity: https://www.youtube.com/watch?v=XOjd_qU2Ido&t=810s, consultado 28/04/2023

YouTube – Unity Tutorial: Metroidvana style level changing (scene changing): https://www.youtube.com/watch?v=r37YoRWYjX0, consultado 23/04/2023

YouTube – Upgrade, Power-up, Ability Unlock System – 2D Platformer Metroid in Unity 2021 [Part #5]: https://www.youtube.com/watch?v=9n5PPSitg_0, consultado 09/05/2023

Anexos

Anexo A: Glosario

- **Metroidvania:** Género de videojuegos que tiene elementos de exploración, plataformas, acción…
- **Pixel-art:** Forma de arte digital donde se editan las imágenes a nivel de píxel. Muy utilizado en videojuegos antiguos, en la última época ha vuelto a ganar popularidad entre los juegos independientes.
- **Indie:** Abreviación de independiente, hace referencia a los juegos desarrollados por individuos o pequeñas desarrolladoras y no por las grandes compañías.
- **Scroll lateral:** Método de desplazamiento donde el jugador se mueve por el escenario a la izquierda o a la derecha.
- **Power-up:** Mejora o habilidad que obtiene el jugador, muy habitual en los *metroidvania.*
- **Jefe de nivel:** El enemigo más poderoso de un nivel o zona, que el jugador debe derrotar para avanzar en el juego.
- **Backtracking:** Técnica de diseño de nivel en la que se hace al jugador volver sobre sus pasos para avanzar en la historia o conseguir secretos. Es típico que el jugador vea indicios de esto al jugar y recuerde estas pistas para volver a explorar las zonas ya visitadas.
-

Anexo B: Entregables del proyecto

- Informe del proyecto: Descripción del proyecto realizado, la propuesta, conceptos y diseños.
- Ejecutable de Tower Shift: Archivo ejecutable del juego.
- Tráiler del juego: Video breve para anunciar el juego.
- Video de presentación: Video defensa del proyecto en el que se explica en que consiste y como se ha llevado a cabo.

Anexo C: Currículum Vitae

IÑIGO SOBA JIMÉNEZ

Paseo Donantes de Sangre Portal 2
ESC 8 2A , 31015, Pamplona
Española
654087948
inigosoba@gmail.com

RESUMEN PROFESIONAL

Estudiante de diseño de videojuegos, con gran interés en acceder al mercado laboral para poner en práctica mis conocimientos. Soy una persona trabajadora, comunicativa y comprometida. Desearía encontrar una oportunidad para adquirir más experiencia profesional.

FORMACIÓN

Universidad Pública de Navarra, *Pamplona, 2020*
Grado en Ingeniería Informática: Ingeniería del Software

APTITUDES

- Flexibilidad
- Trabajo en equipo
- Adaptabilidad
- Motivación

IDIOMAS

Español: Idioma nativo

Inglés: B2
Intermedio alto

Euskera: B2
Intermedio alto