

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA**

**GRADO EN DISEÑO Y DESARROLLO DE VIDEOJUEGOS**

**Curso Académico 2021/2022**

**Trabajo Fin de Grado**

**Slime Journey**

Diseño y desarrollo de un videojuego de plataformas

2D con elementos metroidvania en Unity.

**Autor:** Manuel Rodríguez Bonete

**Directores**:

Mario Manzano Vázquez

Daniel Burón García

Agradecimientos

Esto va a ser bastante complicado la verdad

* Mama papa y hermanas
* Viejos lesbianos
* Exiled
* Sonia y Emilio
* Mario

*Resumen*

A lo largo del documento se irán abordando todas las cuestiones acerca del desarrollo de un producto software como es un videojuego, desde el estado del arte y el estudio hecho previamente sobre los géneros base sobre los que se sustenta este proyecto, pasando por todo el trabajo realizado y se terminará hablando de los resultados obtenidos y un post-mortem.

Este proyecto ha sido realizado de manera conjunta con *Miguel Ferrer García* en el que mi compañero se ha encargado de toda la realización de los efectos visuales, assets, fondos y shaders que hay en el videojuego.

Durante todo el desarrollo se han utilizado herramientas distintas para el correcto funcionamiento y persistencia de los datos, tales como Github Desktop para control de versiones, Word para el desarrollo de toda la documentación y GDD’s, Unity 2020.18f para el desarrollo del videojuego y Visual Studio para la realización de código.

Por mi parte se ha desarrollado toda la parte de programación, haciendo gran énfasis en el movimiento del personaje, crear elementos disfrutables dentro del escenario que interactúen de formas distintas con el jugador y todos los sistemas internos para que todo funcione internamente de forma correcta y sin errores.

Aparte de toda la parte de programación mencionada anteriormente, otra de mis tareas dentro del desarrollo de este proyecto ha sido la concepción de la historia del videojuego, de todas las mecánicas de personaje y de mundo, así como el diseño de todos los niveles de los que dispone el videojuego.

Por último, el diseño de la interfaz, así como la programación de esta misma y el diseño de sonido también han sido diseñados por mí.

Términos clave

* Unity
* Videojuego
* Plataformas
* Metroidvania
* Alta dificultad

Contenido

[1. Preproducción 10](#_Toc138269448)

[1. Brainstorming 10](#_Toc138269449)

[2. Estado del arte 10](#_Toc138269450)

[3. GDD inicial – Punto de partida 10](#_Toc138269451)

[4. Herramientas 10](#_Toc138269452)

[5. Planificación y fases 10](#_Toc138269453)

[1. Alfa 10](#_Toc138269454)

[2. Beta 11](#_Toc138269455)

[3. Versión candidata 11](#_Toc138269456)

[4. Gold Máster 11](#_Toc138269457)

[2. Game Design Document (GDD) 12](#_Toc138269458)

[1. Elementos base del videojuego 12](#_Toc138269459)

[1. Introducción 12](#_Toc138269460)

[2. Conceptos y objetivos 13](#_Toc138269461)

[3. Pilares del diseño 15](#_Toc138269462)

[4. Temática e historia 15](#_Toc138269463)

[2. Elementos de la mecánica del videojuego 16](#_Toc138269464)

[1. Mecánicas del personaje 16](#_Toc138269465)

[2. Mecánicas del mundo 16](#_Toc138269466)

[3. Elementos de diseño del videojuego 20](#_Toc138269467)

[1. Personajes 20](#_Toc138269468)

[2. Zonas del mundo 20](#_Toc138269469)

[3. Apartado visual 20](#_Toc138269470)

[4. Audio y sonido 27](#_Toc138269471)

[5. Interfaz 27](#_Toc138269472)

[6. Coleccionables y diálogo 27](#_Toc138269473)

[3. Resultados obtenidos 28](#_Toc138269474)

[1. Flujo del videojuego 28](#_Toc138269475)

[2. Niveles 28](#_Toc138269476)

[3. Post - mortem 28](#_Toc138269477)

[4. Bibliografía 29](#_Toc138269478)

Tabla de ilustraciones

[Ilustración 1 - Mapa nivel 1 Super Meat Boy 6](file:///C:\Users\Usr\Documents\GitHub\Tfg\Assets\Documentos\MemoriaTFG.docx#_Toc105235967)

[Ilustración 2 - Mapa continuo Hollow Knight 6](file:///C:\Users\Usr\Documents\GitHub\Tfg\Assets\Documentos\MemoriaTFG.docx#_Toc105235968)

[Ilustración 3 - Dificultad y progresión 7](file:///C:\Users\Usr\Documents\GitHub\Tfg\Assets\Documentos\MemoriaTFG.docx#_Toc105235969)

[Ilustración 4 - Ejemplo de enemigo 11](file:///C:\Users\Usr\Documents\GitHub\Tfg\Assets\Documentos\MemoriaTFG.docx#_Toc105235970)

[Ilustración 5 - Jefe de tipo Persecución 12](#_Toc105235971)

[Ilustración 6 - Primera paleta creada 13](file:///C:\Users\Usr\Documents\GitHub\Tfg\Assets\Documentos\MemoriaTFG.docx#_Toc105235972)

[Ilustración 7 – Plataformas básicas 14](#_Toc105235973)

[Ilustración 8 - Paleta de cajas 14](file:///C:\Users\Usr\Documents\GitHub\Tfg\Assets\Documentos\MemoriaTFG.docx#_Toc105235974)

[Ilustración 9 - Fondos y decoraciones 15](file:///C:\Users\Usr\Documents\GitHub\Tfg\Assets\Documentos\MemoriaTFG.docx#_Toc105235975)

[Ilustración 10 - Decoraciones maquinaria 15](file:///C:\Users\Usr\Documents\GitHub\Tfg\Assets\Documentos\MemoriaTFG.docx#_Toc105235976)

[Ilustración 11 - Paleta nevada 16](#_Toc105235977)

[Ilustración 12 - Enemigo Mosca 17](#_Toc105235978)

[Ilustración 13 - Enemigo Spinner 17](#_Toc105235979)

[Ilustración 14 - Enemigo Araña 17](#_Toc105235980)

[Ilustración 15 - Enemigo Fantasma 17](#_Toc105235981)

[Ilustración 16 - Enemigo Big Snake 17](#_Toc105235982)

# Preproducción

## Brainstorming

## Estado del arte

## GDD inicial – Punto de partida

Documento Anexo nº1 – GDD\_Inicial

## Herramientas

Unity

Visual

Paint.net

Trello

## Planificación y fases

Si tomamos un videojuego como un producto software que se va a comercializar y a explotar, se deben diferenciar una serie de fases cada una con una serie de objetivos y fechas. Todo producto software debe pasar por una serie de iteraciones para llegar a una forma definitiva comercializable y satisfactoria, por ello, dejaremos de lado un enfoque de desarrollo en cascada y se enfocará como un desarrollo ágil de software (videojuego, en este caso).

Cuando hablamos de desarrollo con metodologías ágiles, hablamos de pequeños pasos iterativos e incrementales, donde en cada fase del proceso habrá una pequeña planificación de qué se va a desarrollar para esa misma fase, análisis de requisitos para poder desarrollar lo anteriormente mencionado, diseño o modificación de las mecánicas anteriormente mencionadas (más bien modificación, si es que la hubiera) y una pequeña fase final de testing, tanto interno como externo.

### Alfa

Esta primera versión alfa, será una pequeña demo jugable del videojuego, donde el usuario podrá probar las mecánicas “core” del videojuego. En este punto del desarrollo, ya tenemos un GDD inicial creado, un documento que recoge todos los puntos a tratar y desarrollar del videojuego. Esta primera fase dista mucho del resultado final del producto, pero es una fase muy importante para comprobar que las mecánicas diseñadas en el GDD son viables de seguir desarrollando.

En esta primera iteración, Across The Ground se subirá a Itch.io, acompañado de un pequeño cuestionario acerca de qué les ha parecido el juego a los usuarios que lo han probado.

La alfa se puso a disposición del público general el día 12 de junio.

### Beta

La fase beta es la 2ª iteración del producto final que se desarrolla. En este punto, las mecánicas generales del videojuego deben estar al completo, o casi en su punto final, los elementos visuales deben ser los definitivos, aunque pueden estar sujetos a cambio, el sonido, música de ambiente y otros efectos sonoros deben estar bien pulidos y el juego debe estar carente de bugs.

El objetivo de esta fase beta, no es otro que conseguir un producto lo más pulido posible, reportar la mayor cantidad de bugs y si esta fase se mostrase al público (que así es), recibir feedback sobre el producto.

La versión beta fue publicada el X de X.

### Versión candidata

Una versión candidata de un videojuego, es una versión del producto que está finalmente lista para comercializar abiertamente.

En este punto del desarrollo, el videojuego se puede decir que está terminado, solo a falta de un buen testeo a todos los niveles del juego para erradicar la mayor parte de bugs que sea posible antes del lanzamiento oficial.

No tiene porqué haber una única versión candidata, estas versiones se enumeran como RC1, RC2, dependiendo del número de iteraciones que se haya hecho, hasta llegar a un nivel de pulido lo suficiente bueno como para comercializarlo.

### Gold Máster

La versión Gold Máster es la última versión del videojuego antes de pasar a la postproducción, tema que no abordaremos aquí.

Una versión Gold es la que se comercializa a día 1, una versión completa del videojuego, con la mayor parte de los bugs eliminados y el contenido completo del videojuego.

# Game Design Document (GDD)

## Elementos base del videojuego

### Introducción

El concepto y objetivo de este proyecto de fin de grado es el diseño y desarrollo de un videojuego del género plataformas, concretamente en un estilo 2D, con una serie de elementos característicos del género de los metroidvania.

Esto consiste, en resumen, en dotar al jugador de una progresión de su personaje a lo largo de la aventura y a medida que este avance, se desbloqueen zonas paralelas y secundarias del mapa donde el jugador tenga nuevo contenido. Este concepto se concretará más adelante.

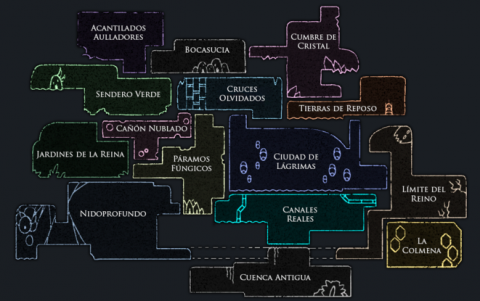
Dentro del género de plataformas 2D, se pueden encontrar videojuegos como “Super Mario Bros”, “Celeste”, “The End is Nigh” o “Super Meat Boy”, hay una gran diferenciación entre ellos, variando así mismo sus mecánicas básicas, su nivel de progresión, así como la distribución de los niveles. En Celeste podremos encontrar niveles amplios separados por un mapa por zonas, como en Super Mario Bros, pero, al contrario, The End is Nigh, opta por unificar todo el mundo sin separarlo en zonas inconexas, a fin de dotar al videojuego de un dinamismo del cual, los otros carecen, ya que cuando el jugador termina un nivel, debe volver al mapa del mundo y seleccionar el siguiente nivel. Otro punto por enfocar dentro del proyecto es el tema de la dificultad, los videojuegos de plataforma tienen una dificultad, por lo general, bastante alta, dependiendo del enfoque se le dé. En Super Mario Bros, la dificultad que se le da al videojuego no puede ser excesivamente alta, ya que está enfocado para un público joven e inexperto, por otra parte, si un jugador de plataformas busca un desafío, puede optar por títulos como Super Meat Boy, Celeste o The end is nigh, donde la dificultad del juego es uno de los pilares que lo soportan. En cualquier de estos tres juegos, el jugador morirá al instante si un enemigo u obstáculo del mapa le golpea.

Ilustración 1 - Mapa nivel 1 Super Meat Boy

Ilustración 2 - Mapa continuo Hollow Knight

Dentro del género metroidvania, podemos destacar referentes como “Hollow Knight”, “Blasphemous” o “Axiom Verge”, con distintas temáticas, mecánicas a la hora de afrontar los desafíos que le propone al jugador y maneras que disciernen a la hora de mostrar y otorgar los elementos que caracterizan a este género concreto. En Hollow Knight, cuando el jugador consigue una mejora, esta se adhiere a nuestro personaje y hasta el fin de su aventura, el caballero hueco podrá utilizar esa nueva habilidad, tales como la habilidad para realizar un doble salto con las alas de monarca o el poder realizar un avance rápido con la capa de polilla, casi al inicio del videojuego. Por otra parte, Blasphemous implementa una mecánica de equipamiento de habilidades, donde el jugador tendrá que buscar una “Prie Dieu” o una hoguera al más puro estilo Dark Souls, para poder intercambiar estas habilidades y acceder a lugares a los que previamente, sin esas habilidades, no puedes.

Este tipo de mecánicas añaden rejugabilidad y son una forma de incentivar al jugador a explorar zonas nuevas.

Con esta breve introducción sobre los géneros sobre los cuales se basarán este videojuego, ya podemos comenzar con la explicación y concreción de los elementos que compondrán el título a desarrollar.

### Conceptos y objetivos

El objetivo básico de este proyecto, como se ha mencionado anteriormente, es desarrollar un videojuego de plataformas 2D con elementos metroidvania, esto plantea una división muy clara de los retos que nos encontraremos a lo largo de este proyecto, los retos de diseño y los retos de desarrollo.

Diagrama

Descripción generada automáticamenteEn un inicio, los principales problemas en el diseño del videojuego que se pueden encontrar son algunos de los siguientes; el diseño de sus mecánicas principales como videojuego de plataformas, sus mecánicas secundarias como videojuego metroidvania, el diseño de un mapa lo suficientemente grande e interesante como para que el jugador tenga interés a la hora de explorarlo, de investigar por los recovecos más oscuros y desenmascarar todo lo que el título le quiere ofrecer. La dificultad del juego es otro elemento para tener muy en cuenta en el diseño del videojuego, se deberá enseñar al jugador a sortear los obstáculos más básicos y cederle una progresión adecuada para que no se sienta abrumado, pero no se aburra en ningún momento.

Ilustración 3 - Dificultad y progresión

Por otra parte, nos encontraremos con los problemas del desarrollo del videojuego, es decir, su parte más técnica, las tripas del videojuego.

En primer lugar, el movimiento del personaje y sus mecánicas “core” (núcleo, es decir, principales) es el pilar que sostendrá todo el videojuego, si este movimiento y derivados no es disfrutable, fluido y, en esencia, divertido, un videojuego de este estilo está abocado al fracaso, por lo que el principal objetivo que nos encontramos desde el minuto uno, es desarrollar un controlador de movimiento para nuestro personaje que sea fluido y disfrutable, pues la mecánica principal del videojuego es, en esencia, moverse y saltar.

El siguiente reto principal que nos encontraremos, será la gestión de escenas, la gestión de todo el mapa, cómo conseguir el máximo rendimiento y tiempos de espera mínimos para no aburrir al jugador con tiempos de carga infinitos, pero, por otra parte, no tener pantallas de carga cada 30 segundos. Este problema se evaluará más adelante y se estudiará la forma de maximizar el rendimiento y minimizar tiempos de carga.

A medida que el usuario avance por el mundo virtual en el que se encuentra, el viajar de un punto determinado “A” hacia un punto determinado “B”, puede llegar a ser tedioso, sobre todo si estos puntos determinados se encuentran muy separados entre sí, por lo que se desea implementar un sistema de viaje rápido entre ciertos puntos del mapa, los cuales el jugador deberá descubrir a medida que progrese en su aventura.

Por la temática del videojuego, que se tratará en el siguiente apartado, se ha pensado un sistema de viaje rápido por túneles, donde, temáticamente hablando, estos puntos determinados se conectarán por medio de una serie de entramados túneles subterráneos. En todos los videojuegos, cuando el jugador viaja por medio de teletransportes o viaje rápido, siempre hay una pantalla de carga que provocará una pérdida de tiempo del jugador, por ello, se ha pensado en desarrollar un mini juego intermedio, generado procedimentalmente, de una forma semi-aleatoria, para otorgar algo de dinamismo al transporte rápido.

Todo videojuego, tiene objetos con los que interactuar, por lo que, la interacción con los objetos y la gestión de todo lo que esto conlleva, será otro de los retos que se deberán asumir, conseguir realizar distintos controladores que lleven a cabo tareas de forma simultánea para un funcionamiento correcto, así como el guardado y cargado de toda la información de nuestro personaje y de los objetos que ha conseguido, para poder tener varias partidas de forma simultánea sin tener que borrar los datos. Esto se puede realizar serializando la información que se requiera de nuestro personaje y del estado del mundo, para así guardarlo en un archivo “json” y posteriormente poder cargar ese archivo de datos para poder continuar con la aventura exactamente donde el jugador la dejó anteriormente.

Este punto, tiene varias formas de enfocarlo, guardado automático cada cierto tiempo, para que el jugador no tenga que preocuparse de guardar de forma manual, guardar la partida en ciertos puntos, como las hogueras en Dark Souls o un guardado más clásico desde el menú como se hacía antiguamente. Estos métodos se valorarán posteriormente.

### Pilares del diseño

Hablamos de los pilares que sustentan el videojogo

### Temática e historia

El videojuego tratará sobre una civilización formada por seres no humanoides, que habitan un planeta que parece ser la tierra, pero donde no hay humanos.

Al comienzo de la historia, al personaje principal se le encomendará la misión de ir a visitar el centro de la Tierra para conseguir revelar la razón por la cual el núcleo del planeta se está deteriorando y, por ende, matando el hogar de esta sociedad poco a poco.

Para acceder al centro de la Tierra, el protagonista deberá lanzarse por un volcán que lleva poco tiempo apagado, debido a la baja intensidad magmática del núcleo terrestre.

Cuando el personaje llega al fondo del volcán, este se encuentra con unas ruinas un tanto extrañas, con formas, dibujos y objetos que no consigue reconocer. Tras avanzar por este nuevo mundo subterráneo que ha encontrado, descubre que, bajo el volcán, existe una sociedad que lleva siglos subsistiendo a base de robar energía del núcleo de la tierra.

## Elementos de la mecánica del videojuego

### Game Loop

### Controles

### Personaje

Posibles mecánicas que el personaje tendrá de inicio:

* Saltar
* Andar
* Correr
* Agacharse
* Agarrarse a las esquinas

Posibles mecánicas que el personaje irá adquiriendo a lo largo de la aventura:

* Doble salto
* Dash
* Romper muros
* Trepar

### Escenario

#### Distribución del mundo

Para la concepción del mundo, se busca un mundo continuo, donde cualquier jugador pueda llegar del punto A al punto B sin necesidad de un teletransporte o pasar por medio de una pantalla de carga. Aún con este formato de mapa continuo, se habilitará un sistema de viaje rápido para facilitar el progreso del jugador.

El mundo se dividirá por zonas, donde cada nuevo terreno, tendrá una ambientación única y añadirá alguna nueva mecánica, que incrementará alguna ya presentada anteriormente o presentará un nuevo desafío al jugador.

Estas mecánicas, podrían ser, nuevos enemigos con nuevos patrones de movimiento o patrulla, plataformas móviles, plataformas que se rompen con contacto y reaparecen, zonas con menos gravedad, zonas acuáticas con nuevo set de movimientos…

A medida que el jugador descubra nuevas zonas más avanzadas, su dificultad aumentará, ya que no tendrá que lidiar con las mecánicas nuevas que esa zona le presente, sino con todas las que ha enfrentado con anterioridad, ocasionando un progreso exponencial en la dificultad.

Aunque el avance en la dificultad sea cada vez mayor, también se tendrá presente, que el jugador cada vez jugará mejor y dominará fácilmente las mecánicas que lleven más tiempo presentes en su partida, por lo que solo tendrá que aprender a dominar las nuevas mecánicas que encuentre.

Cada mundo temático del videojuego, estará compuesto por un conjunto de salas, donde cada sala tendrá una o varias entradas, así como una o varias salidas, proporcionando varios caminos al jugador.

El punto por el que el jugador haya accedido a la sala, será el punto de respawn de ese nivel.

#### Elementos metroidvania

A lo largo de la aventura, el jugador irá encontrando una serie de obstáculos que a priori parecen ser insorteables, estos pueden ser, plataformas muy altas a las que no se puede llegar sin el doble salto, muros verticales altos que sin la habilidad para trepar o escalar no son accesibles, muros agrietados, que posteriormente se podrán destruir con la habilidad de cargar.

Estos puntos donde el jugador podrá volver cuando consiga la habilidad requerida para poder sortear cierto obstáculo, abrirá nuevas zonas del mapa, más o menos amplias, con nuevos desafíos, nuevos niveles y nuevas recompensas.

#### Sistema de reaparición

Dependiendo del enfoque se les otorgue a los videojuegos del género de plataformas, el sistema de vidas puede diseñarse y desarrollarse de una forma u otra, y, en este caso, se busca un sistema de vidas similar al de “The End is Nigh”, donde el jugador como tal, no poseerá un número de vidas o de golpes, sino que morirá de cualquier golpe, siendo teletransportado al punto de respawn(punto de reaparición) de la sala donde se encuentra. Este punto de respawn será determinado por el punto de entrada a la sala, siendo el mismo hasta que consiga cambiar de sala o encuentre un punto de respawn (si es que lo hubiera en esa sala).

Este sistema de reaparición, está diseñado de esta forma para que las salas sean cortas de duración, pero complejas (hasta cierto punto) de completar, haciendo que el jugador muera varias veces hasta que consiga finalizar esa sala y avanzar a la siguiente, consiguiendo un ciclo de vida y muerte permanente.

#### Sistema de viaje rápido

Para que el jugador no tenga que realizar grandes viajes por el mundo, se ha diseñado un sistema de viaje rápido por puntos concretos. Es decir, en determinadas zonas del mapa, el personaje principal podrá desbloquear el viaje rápido a otras zonas ya descubiertas.

Este sistema de viaje rápido es similar al de Hollow Knight, donde para poder utilizar los caminos de ciervos, tienes que desbloquearlos previamente, en el caso de Hollow Knight, con monedas. En el momento en el que tienes un camino desbloqueado, el jugador puede viajar entre estos puntos libremente.

Se ha pensado un sistema de mini juego entre los puntos de viaje rápido, donde con una subrutina cargará una escena muy ligera en el que el jugador tendrá que sortear un pequeño laberinto generado procedimentalmente, proporcionando un valor extra al viaje rápido y un pequeño entretenimiento para el jugador, en vez de una pantalla de carga.

#### Sistema de estadísticas

Estadísticas

#### Sistema de coleccionables

* Coleccionables por cada pantalla
* Coleccionables de texto
* Skills de jugador

#### Inventario y Objeto de inventario

Todo el inventario, justo después de los coleccionables

#### Cambio de nivel

* Camera points
* Camara
* Limite de escenas

#### Enemigos

Este videojuego no está pensado como un videojuego con combate, como podemos encontrar en Ori and the Blind forest o Hollow Knight, aunque no son su mecánica principal, sino que los enemigos son un añadido a la dificultad que actuarán como entes con una IA muy simple.

Ilustración 4 - Ejemplo de enemigo

Esta IA consistirá en dotar al enemigo de un camino predeterminado y que generalmente, ocasionará la muerte del personaje principal de un único golpe.

Los “Goomba” de Mario Bros, son un ejemplo de este tipo de enemigos.

#### Jefes

Los jefes están determinados como desafíos extraordinarios cada cierto tiempo, por ello, en un principio, habrá 1 jefe por cada zona del mundo.

Estos jefes serán enfrentamientos excepcionales, donde se pondrá a prueba al jugador mediante las mecánicas que ha ido aprendiendo a lo largo de la zona en la que se encuentre. Este tipo de enfrentamientos está pensado como salas más grandes, donde se deba atravesar un determinado camino en un tiempo corto, zonas de persecución, como en “Super Meat Boy” o zonas donde habrá que perseguir al jefe por unos obstáculos, como en “Super Mario Bros 3D Land”, donde Bowser te lanza objetos que tienes que esquivar hasta que consigues derrotarle al final del nivel.



Ilustración 5 - Jefe de tipo Persecución

#### Objetos interactuables

Describimos todas las plataformas y elementos desarrollados

#### Gestión de escenas

#### Persistencia de datos y perfiles

### Interfaz de usuario

## Elementos de diseño del videojuego

### Personajes

#### Destra

#### Personajes sin consciencia

#### ¿Aliens?

### Zonas del mundo

#### Bosque primigenio

#### Bosque 2º

#### Minas 1

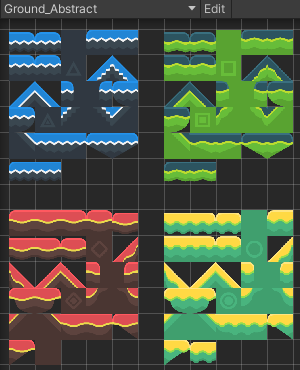
#### Minas profundas

### Apartado visual

#### Personaje principal

Ponemos imágenes del perosnaje principal, animaciones, etc ciñéndonos al apartado grafico.

#### TileMaps

Para el diseño de los niveles de Across the Ground, se ha optado por utilizar una herramienta propia de Unity, los TileMaps.

Un TileMap en Unity es un GameObject con un componente que permite dibujar tiles (o celdas) en el escenario mediante una serie de paletas que el propio diseñador crea, mediante spritesheets o mediante un conjunto de imágenes.

Estos TileMaps permiten al diseñador crear una amplia gama de niveles, diseños y mapas con solo unos pocos sprites.

Ilustración 6 - Primera paleta creada

Estos TileMaps se agrupan dentro de un GameObject llamado Grid, que es el objeto que determina el tamaño de las celdas y es el padre de los TileMaps.

Dentro del Grid, los TileMaps se diferenciarán dependiendo del tipo de celda que queramos colocar, es decir, crearemos un GameObject para el suelo, al que le pondremos un componente “Tilemap Collider 2D” para que el jugador pueda colisionar correctamente con ellos. En cambio, si se desea añadir decoración, tiles de fondo o coleccionables, estos se deberán crear en objetos distintos para no dotarles de comportamientos no deseados.

A continuación, se van a mostrar las paletas creadas para el correcto diseño de los niveles:

Esta paleta, denominada “PlatformerTiles” está conformada por varios packs de Assets con un estilo visual muy parecido, pero con diferencias notables, para que haya una gran variabilidad en los suelos de los distintos mundos por el que el jugador irá avanzando.

Esta paleta, sumada a la anteriormente mostrada, serán ambas suficientes para tener una suma de elementos aceptables.

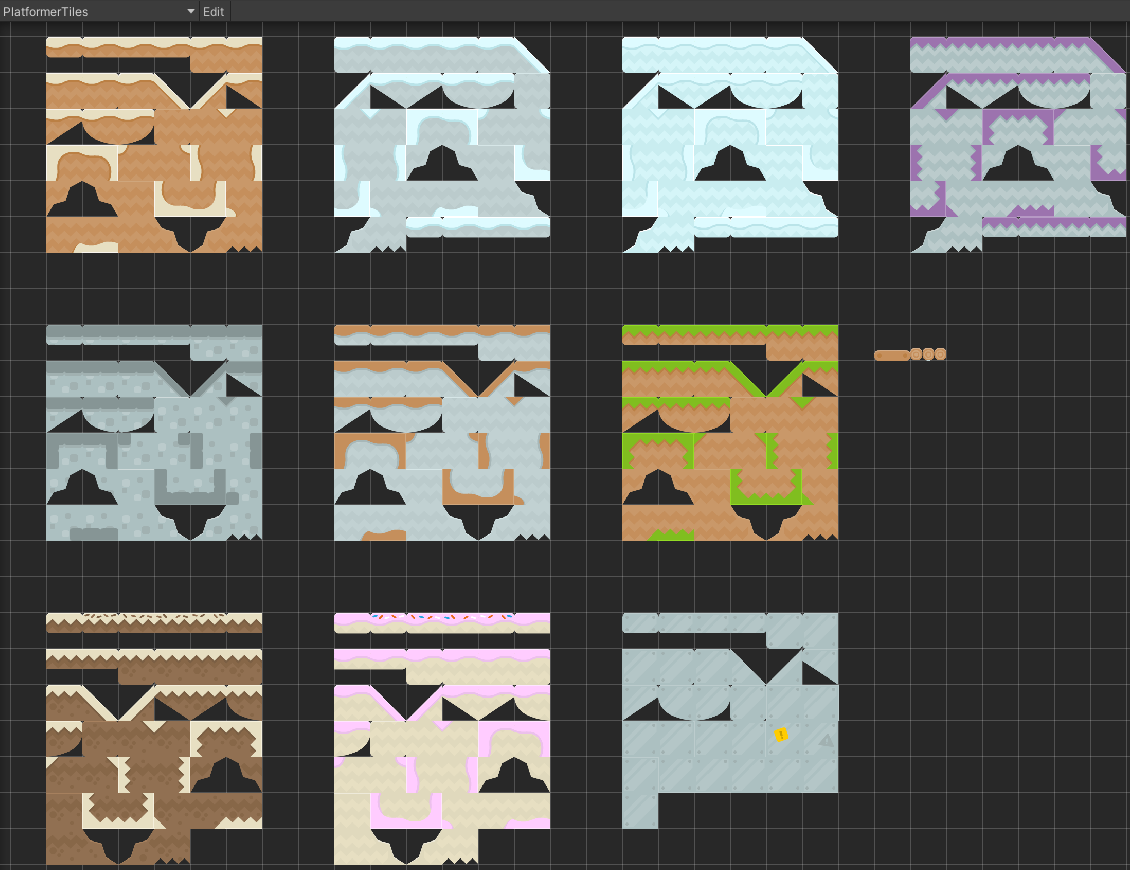


Ilustración 7 – Plataformas básicas



Ilustración 8 - Paleta de cajas

Esta paleta, nombrada como “Boxes” es un conjunto de cajas de un mismo pack, añadidas con el fin de añadir elementos de fondo visibles y con diferentes colores.

Este conjunto de tiles, con título “Backgrounds” ha sido creada con el motivo de tener unos fondos y decoraciones extras.

Ilustración 9 - Fondos y decoraciones

Ilustración 10 - Decoraciones maquinaria

Este conjunto denominado “RequestExp” dota al videojuego de unas decoraciones con maquinaria, trampas y algunos mecanismos.

Por último, este conjunto aporta una serie de elementos de nieve, para añadir ciertas decoraciones a los niveles gélidos del mundo.



Ilustración 11 - Paleta nevada

#### Fondos de niveles

Sadad

#### Enemigos

Los enemigos de Across The Ground son visualmente muy sencillos, con pequeñas animaciones para hacerlos algo más vistosos.

Todos los enemigos del juego se han convertido en prefabs para poder distribuirlos de la forma requerida a través de los niveles, simplificando la complejidad del videojuego.

Los enemigos que el jugador puede encontrar son los siguientes:

* Moscas:

Ilustración 12 - Enemigo Mosca

* Spinners:

Ilustración 13 - Enemigo Spinner

* Arañas:

Ilustración 14 - Enemigo Araña

* Fantasma:

Ilustración 15 - Enemigo Fantasma

* Big Snake:

Ilustración 16 - Enemigo Big Snake

* Caracol:

Ilustración 17 - Enemigo Caracol

* Murciélago



Ilustración 18 - Enemigo murciélago

* Rana



Ilustración 19 - Enemigo Rana

* Slimes

Ilustración 20 - Enemigos Slime

### Audio y sonido

### Interfaz

### Coleccionables y diálogo

# Resultados obtenidos

## Flujo del videojuego

## Niveles

## Post - mortem

# Bibliografía

Assets:

https://kenney.nl/assets/simplified-platformer-pack

https://kenney.nl/assets/foliage-pack

https://kenney.nl/assets/platformer-pack-industrial

https://kenney.nl/assets/platformer-pack-medieval

https://kenney.nl/assets/jumper-pack

https://kenney.nl/assets/background-elements

https://kenney.nl/assets/platformer-art-deluxe

https://kenney.nl/assets/platformer-art-buildings

Unity tutorials:

<https://www.youtube.com/watch?v=XhwRYNie-aI&t=1s&ab_channel=RehopeGames>

<https://www.youtube.com/watch?v=sfDnN-Im7rY&ab_channel=RehopeGames>

<https://www.youtube.com/watch?v=KbtcEVCM7bw&ab_channel=DawnosaurStudios>

<https://www.youtube.com/watch?v=TTKPmPvekUY&ab_channel=TacticalProgrammer>

<https://www.youtube.com/watch?v=A_F8R3eGtrs&ab_channel=TacticalProgrammer>

<https://www.youtube.com/watch?v=JIASeoOU274&ab_channel=TacticalProgrammer>

<https://www.youtube.com/watch?v=Do4LdlKB0bY&ab_channel=TacticalProgrammer>

<https://www.youtube.com/watch?v=1bqEGHyR-ao&ab_channel=TacticalProgrammer>

<https://www.youtube.com/watch?v=ryISV_nH8qw&ab_channel=Brackeys>

<https://medium.com/nerd-for-tech/animated-tilemaps-for-unity-2020-a659dbc915bc>

Metodología y planificación

<https://www.linkedin.com/pulse/game-development-life-cycle-sumit-jain/>

<https://medium.com/wharf-street-studios/game-development-a-play-into-the-process-494fb7b0ecbe>

<https://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software>