

Fractal Runaway

Autor: Ricardo Villegas Ortiz
Tutor: Xavier Pallicera Malivern
Profesor: Joan Arnedo Moreno

Grado en Ingeniería informática
Itinerario de Computación

18/06/2023



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

FICHA DEL TRABAJO FINAL

| | |
|---|--|
| Título del trabajo: | <i>Fractal Runaway</i> |
| Nombre del autor: | <i>Ricardo Villegas Ortiz</i> |
| Nombre del colaborador/a docente : | <i>Xavier Pallicera Malivern</i> |
| Nombre del PRA: | <i>Joan Arnedo Moreno</i> |
| Fecha de entrega (mm/aaaa): | <i>06/2023</i> |
| Titulación o programa: | <i>Grado en Ingeniería informática</i> |
| Área del Trabajo Final: | <i>Videjuegos</i> |
| Idioma del trabajo: | <i>Español</i> |
| Palabras clave | <i>Videjuegos, Plataformas, generación de escenarios, Autómata Celular, Curvas fractales</i> |
| Resumen del Trabajo: | |
| <p>Fractal Runaway es un juego de plataformas que he decidido llevar a cabo como mi TF. Con este proyecto, espero poder adquirir nuevos conocimientos relativos a la creación y diseño de un videojuego con tal de poder encaminar mi itinerario académico a un futuro máster en desarrollo y programación de Videjuegos.</p> <p>Teniendo en cuenta la duración del proyecto, he decidido crear una breve historia que hará de hilo conductor del videojuego, que dará lugar a los diferentes niveles que el jugador deberá ir superando mediante las diferentes herramientas proporcionadas.</p> <p>He considerado interesante la creación de los diferentes niveles mediante un algoritmo de generación procedural realizado con un autómata celular. Además, para que haya una estructura básica en los niveles, implementaré diferentes curvas fractales, que permitirán que haya unas pautas para que el escenario pueda tener sentido dentro de lo deseado para mi proyecto a la hora de generar el 'mundo'.</p> <p>Mi producto final incluirá la generación de escenario, diferentes enemigos que añadan cierta dificultad y objetos que deberás recolectar antes de poder pasar al siguiente nivel.</p> <p>Con esto, pretendo crear un videojuego que proponga un reto, gracias a la variabilidad del escenario de juego y los diferentes enemigos en el recorrido.</p> | |

Dedicatoria/Cita

Este trabajo está dedicado a Anna, que me ha motivado a continuar estudiando y creciendo como persona.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a David, mi pareja, por darme su constante apoyo y ánimos durante el proyecto.

Quiero también agradecer a Alex y Leni, que me han ayudado con el diseño de los personajes y de las pistas de audio.

Abstract

Fractal Runaway is a platform game that I have decided to carry out as my final project. With this project, I hope to acquire new knowledge related to the creation and design of a video game to be able to start my academic path toward a future master's degree in video game development and programming.

Considering the duration of the project, I have decided to create a short story that will be the main thread of the video game, which will lead to the different levels that the player will have to overcome by means of the different tools provided.

I have found it interesting to create the different levels by means of a procedural generation algorithm made with a cellular automaton. In addition, to have a basic structure in the levels, I will implement different fractal curves, which will allow there to be some guidelines so that the scenario can make sense within the desired for my project when generating the 'world'.

My final product will include scenario generation, different enemies that add some difficulty, and items that you will have to collect before you can move on to the next level.

With this, I intend to create a challenging video game, thanks to the variability of the game scenario and the different enemies along the way.

Keywords: Videogames, Platforms, Scenario generation, Cellular Automata, Fractal Curves

Índice

| | |
|---|-----------|
| 1. Introducción..... | 11 |
| 1.1. Prefacio..... | 11 |
| 1.2. Descripción/Definición | 12 |
| 1.3. Objetivos generales | 13 |
| 1.3.1. Objetivos principales..... | 13 |
| 1.3.2. Objetivos secundarios | 13 |
| 1.4. Metodología y proceso de trabajo..... | 14 |
| 1.5. Planificación..... | 15 |
| 1.6. Presupuesto | 19 |
| 1.7. Estructura del resto del documento | 21 |
| 2. Análisis de mercado | 23 |
| 2.1. Público objetivo y perfiles de usuario | 25 |
| 2.2. Estado del arte | 29 |
| 2.3. Análisis DAFO..... | 36 |
| 3. Propuesta..... | 37 |
| 3.1. Definición de objetivos/especificaciones del producto | 37 |
| 3.2. Modelo de negocio / Estrategia de marketing | 39 |
| 4. Diseño..... | 40 |
| 4.1. Entorno | 40 |
| 4.2. Requisitos técnicos del entorno..... | 40 |
| 4.3. Herramientas utilizadas | 41 |
| 4.4. Assets y recursos..... | 42 |
| 4.5. Esquemas de arquitectura | 45 |
| 4.6. Diseño de niveles | 46 |
| 4.7. Guion / Sinopsis | 54 |
| 5. Implementación..... | 55 |
| 5.1. Requisitos de instalación | 55 |
| 5.2. Instrucciones de instalación..... | 55 |
| 5.3. Instrucciones del juego | 55 |

| | |
|---|-----------|
| 6. Demostración | 57 |
| 6.1. Prototipos | 57 |
| 6.1.1. Prototipos Lo-Fi..... | 57 |
| 6.1.2. Prototipos Hi-Fi | 58 |
| 6.2. Pruebas de usuario | 58 |
| 6.2.1. Pruebas realizadas..... | 58 |
| 6.2.2. Análisis del formulario..... | 59 |
| 7. Conclusiones y líneas de futuro | 62 |
| 7.1. Conclusiones | 62 |
| 7.1.1. Lecciones aprendidas..... | 62 |
| 7.1.2. Objetivos | 62 |
| 7.1.2. Planificación | 63 |
| 7.2. Líneas de futuro..... | 63 |
| 8. Bibliografía..... | 64 |
| Anexos..... | 67 |

Figuras y tablas

Índice de figuras

| | |
|--|----|
| Figura 1: Diagrama de Gantt..... | 17 |
| Figura 2: Distribución del género en los videojuegos 2021. Fuente: Anuario AEVI 2021 | 25 |
| Figura 3: Distribución de la edad en los videojuegos. Fuente: Anuario AEVI 2021 | 26 |
| Figura 4: Dispositivos más utilizados. Fuente: Anuario AEVI 2021 | 27 |
| Figura 5: Ejemplo 1 mapa The Binding of Isaac. (41) | 30 |
| Figura 6: Ejemplo 2 mapa The Binding of Isaac. (40) | 30 |
| Figura 7: Ejemplo 2 mapa Enter the Gungeon. (42)..... | 30 |
| Figura 8: Ejemplo 1 mapa Enter the Gungeon. (42)..... | 30 |
| Figura 9: Vecindario de Von Neumann (izquierda) y vecindario de Moore (derecha). (15)..... | 31 |
| Figura 10: Primeros 3 pasos de la curva fractal de Peano. (43) | 32 |
| Figura 11: Primeros 5 pasos de la curva..... | 32 |
| Figura 12: Primeros 3 pasos de la curva de Moore. (25) | 34 |
| Figura 13: Primeros 6 pasos de la curva de Hilbert. (26)..... | 35 |
| Figura 14: Primeros 3 pasos de la curva de Sierpinski. (15)..... | 35 |
| Figura 15: Análisis DAFO. (28) | 36 |
| Figura 16 : Sprites del protagonista | 42 |
| Figura 17: Enemigos nivel 1..... | 43 |
| Figura 18: Enemigos nivel 2..... | 43 |
| Figura 19: Diseños de las llaves. | 43 |
| Figura 20: Sprite de un punto de salud. | 43 |
| Figura 21: Tile de fondo. | 44 |
| Figura 22: Tiles de plataformas..... | 44 |
| Figura 23: Sprite de salida del nivel. | 44 |
| Figura 24: Esquema de los menús del juego. (36)..... | 45 |
| Figura 25: Esquema de navegación dentro de un nivel. (36)..... | 46 |
| Figura 26: 4 generaciones del nivel 1. | 47 |
| Figura 27: 4 generaciones del nivel 2. | 48 |
| Figura 28: 4 Generaciones nivel 3. | 49 |
| Figura 29: Curva de Moore sobre el nivel 1. | 50 |
| Figura 30: Curva de Peano sobre el nivel 2. | 50 |
| Figura 31: Curva de Sierpinski sobre el tercer nivel..... | 51 |
| Figura 32: Elementos de un nivel..... | 51 |
| Figura 33: Tutorial del juego. | 53 |
| Figura 34: Prototipo Lo-Fi. (36)..... | 57 |

Índice de tablas

| | |
|--|----|
| Tabla 1: Hitos | 18 |
| Tabla 2: Presupuesto de equipo humano | 19 |
| Tabla 3: Presupuesto de software..... | 20 |
| Tabla 4: Requisitos de Unity 2021 | 40 |
| Tabla 5: Características del ordenador empleado..... | 41 |

1.Introducción

1.1. Prefacio

La industria de los videojuegos ha experimentado un gran crecimiento en las últimas décadas, convirtiéndose en una de las formas de entretenimiento más populares hoy en día. Desde su comienzo en la década de los setenta, se han ido lanzando todo tipo de títulos diferentes ya sean creados por grandes compañías que cuentan con un alto número de empleados o títulos más humildes desarrollados completamente por un pequeño grupo de personas.

La capacidad de entretener, emocionar y de conectar con una amplia audiencia ha sido la razón de que esta industria sea una de las más importantes hoy en día. Y es que no solo son una forma de entretenimiento, también son una gran herramienta educativa y una plataforma para narrar infinitas historias. A nivel social, también han significado una gran herramienta, ya que permiten a los diferentes jugadores formar equipo y competir, conectando así a diferentes personas a lo largo de todo el mundo.

A nivel personal, los videojuegos me han acompañado desde que tengo uso de razón, me han servido como entretenimiento y diversión en los buenos momentos y como refugio en los más duros. Esto ha sido la razón principal por la que he decidido afrontar este reto y enfocar mi futuro hacia el mundo de los videojuegos, para poder formar parte de este increíble mundo.

1.2. Descripción/Definición

La descripción del proyecto propuesto seguirá el *framework* MDA (1), que propone un enfoque para el diseño de juegos que considera tres elementos principales: Mecánicas, Dinámicas y Estéticas. El desarrollo consistirá en un videojuego en 2D para PC del género de plataformas ambientado en el espacio de un futuro distópico, y consistirá en escapar de una nave que ha sido invadida por una especie alienígena.

En cuanto a las mecánicas, el jugador controlará al personaje principal y deberá llegar al final del recorrido evitando tanto obstáculos de terreno como a los diferentes enemigos mediante los diferentes controles (correr, saltar, usar un deslizamiento, un gancho o escalar un muro). El objetivo será avanzar por diferentes salas, desde un punto inicial a un punto final, reuniendo una, dos o tres llaves (en función del nivel) que estarán dispersos a lo largo del recorrido, y que son necesarios para pasar al siguiente nivel. El elemento principal y más importante del juego será el propio mapa, ya que este se generará mediante algoritmos de generación procedural, añadiendo variedad y dificultad.

Las dinámicas del juego se centrarán en la movilidad alrededor del mapa. Al emplear técnicas de generación procedural, nunca habrá dos recorridos iguales. Con tal de superar los diferentes niveles, el jugador tendrá que usar de manera inteligente los controles, moviéndose rápido para que no le alcance ningún enemigo. A medida que avanzan los niveles, se incrementará la dificultad ya que habrá diferentes tipos de enemigos, aumentará el número de llaves en cada nivel y variará el tamaño del mapa entre niveles y la curva que lo forma.

En cuanto a la estética del juego, se centrará en el diseño de niveles y personajes con una estética espacial y futurista. Los colores, efectos de sonido y música empleados son importantes para transmitir al jugador un sentimiento de tensión a lo largo del juego, que se intensificará cuando un enemigo le haya detectado.

En resumen, el videojuego se centrará en las mecánicas de habilidad e ingenio del jugador para sortear obstáculos y enemigos, la generación procedural del mapa para añadir variedad y dificultad, y la estética espacial y futurista para crear una atmósfera tensa y opresiva.

1.3. Objetivos generales

1.3.1. Objetivos principales

Objetivos de la aplicación/producto/servicio:

- Crear un videojuego de 3 niveles que permita ver la intención de este y las mecánicas principales.
- Implementar un algoritmo que permita generar escenarios aleatorios entre diferentes partidas pero que mantengan cierto sentido. Esto se realizará mediante:
 - La implementación de un autómata celular que permita la generación aleatoria de escenario para cada ejecución del juego.
 - La implementación, añadida al autómata celular, de una curva fractal en cada nivel, de modo que los escenarios tengan un mínimo recorrido establecido y no sean del todo aleatorios.
- Crear 2 mecánicas de movilidad alrededor del mapa para dar diferentes opciones al jugador a la hora de completar un nivel:
 - Un deslizamiento que permita recorrer una determinada distancia.
 - Un gancho que permita al jugador adherirse a las paredes y techos del mapa para moverse con mayor fluidez.

Objetivos para el cliente/usuario:

- Implementar controles de movimiento que sean amigables y sencillos de utilizar.
- Crear una trama para el juego:
 - Crear una sinopsis del juego, que incluya una descripción de la trama y el personaje principales.
 - Crear una narrativa detallada que incluya la historia principal, el mundo del juego y la motivación del protagonista.

Objetivos personales del autor del TF:

- Aprender nuevas técnicas y herramientas para el desarrollo de videojuegos, además de mejorar las que ya conozco.

1.3.2. Objetivos secundarios

- Crear o encontrar recursos gráficos y de audio que se adapten a la estética del videojuego.

1.4. Metodología y proceso de trabajo

Para mi proyecto llevaré a cabo trabajos en diferentes áreas. Es importante definir las lo mejor posible cuanto antes, ya que así se pueden planificar mejor los diferentes pasos a seguir en el desarrollo. El desarrollo seguirá una metodología *Agile*, con un enfoque *SCRUM*. La metodología *Agile* se refiere a un enfoque ágil, centrado en la adaptabilidad y entregas continuas, mediante la flexibilidad y la respuesta rápida a los cambios. *SCRUM* se refiere, dentro de la metodología *Agile*, a un marco de trabajo en *sprints*, que tienen una duración determinada y al final de la cual se obtiene *feedback* para el siguiente sprint. Debido a la duración del TF y de cómo están repartidas las entregas, considero que es la metodología óptima.

En primer lugar, para el desarrollo del videojuego utilizaré el motor de Unity (2), con la versión 2021.3.19f1.

En cuanto al apartado gráfico, para la primera y segunda entrega se utilizarán materiales gráficos temporales para el escenario y los personajes, de modo que para las primeras dos versiones se puedan ver los controles y mecánicas implementadas, y de cara a la presentación final, se sustituirán por los definitivos.

Para la creación de dicho entorno, personajes, menús y demás se buscarán materiales que concuerden con la temática del juego. En caso de no encontrar materiales que se adapten a lo que se espera transmitir con el videojuego, se crearán desde cero utilizando editores como Photoshop, Illustrator y Adobe Animate, herramientas que permiten crear y animar los diferentes modelos respectivamente.

En cuanto al guion para la historia, se desarrollará mediante Microsoft Word, pudiendo incluirse algunos esquemas para ejemplificar mediante herramientas como Draw.io. Este guion incluirá tanto el hilo argumental principal, como las características del personaje protagonista y los enemigos, y descripción de diferentes escenarios.

Por último, para la creación de las diferentes pistas musicales y elementos de sonido, se utilizará Adobe Audition.

1.5. Planificación

La planificación se dividirá en cinco etapas diferentes: conceptualización, diseño, desarrollo y fase de pruebas.

La conceptualización del videojuego es la primera etapa para tratar, pues es la fase que establece las bases para el resto del proyecto, y nos permite tener una idea clara del juego que se quiere desarrollar antes de empezar con la implementación. En esta etapa, se definirá la trama, así como el contexto en el que sucede el videojuego y la motivación del personaje. Es importante también acotar el alcance del proyecto, por lo cual tiene que ser una historia breve y sencilla con tal de cumplir los tiempos de entrega. Se realizarán bocetos de los personajes, además de los posibles escenarios y enemigos.

Por otro lado, en la fase de diseño se llevarán a cabo tareas que permitan definir la apariencia y el comportamiento del videojuego. Se crearán diagramas de flujo de navegación por los menús, lo que permitirá establecer de forma clara cómo el usuario interactuará con el juego. También se crearán bocetos de los diferentes personajes, lo que ayudará a visualizar cómo serán los personajes en el juego final. El diseño es una fase importante, ya que un buen diseño de menús y navegación puede ahorrar mucho tiempo y esfuerzo en fases posteriores del desarrollo.

El desarrollo, que es el punto más extenso del proyecto, incluirá varias tareas fundamentales:

En primer lugar, se llevará a cabo el desarrollo de los controles de movimiento del personaje principal, los cuales incluirán movimientos laterales, salto, doble salto, salto en pared, escalar paredes, deslizamiento y gancho. Además, se implementará un algoritmo de generación procedural, aplicando un autómatas celular (3) y diferentes curvas fractales (4), lo cual permitirá crear un escenario con una ruta que no sea del todo aleatoria.

Asimismo, se crearán tres niveles completos que implementen todos los controles de movimiento del jugador y diversas entidades enemigas. Cada uno de los niveles será generado con una curva fractal diferente, dando así mayor variedad al escenario generado. Con tal de que el videojuego cuente con una dificultad variable, y teniendo en cuenta que el escenario se genera de manera automática, he decidido que la mejor manera es que, a

medida que se sucedan los niveles, el tamaño de estos y el número de enemigos sea mayor. De este modo, supondrá un reto cada vez más complicado el reunir los fragmentos y llegar al destino final para superar el nivel.

Una vez que se hayan completado estas tareas, se procederá al testeo y corrección de errores, para garantizar el correcto funcionamiento del juego. Esto se hará mediante la interacción de otros usuarios que probarán el videojuego y darán un *feedback* mediante una encuesta. Finalmente, se llevará a cabo la creación de gráficos y sonidos, con el objetivo de dotar al videojuego de una estética atractiva y una experiencia sonora envolvente.

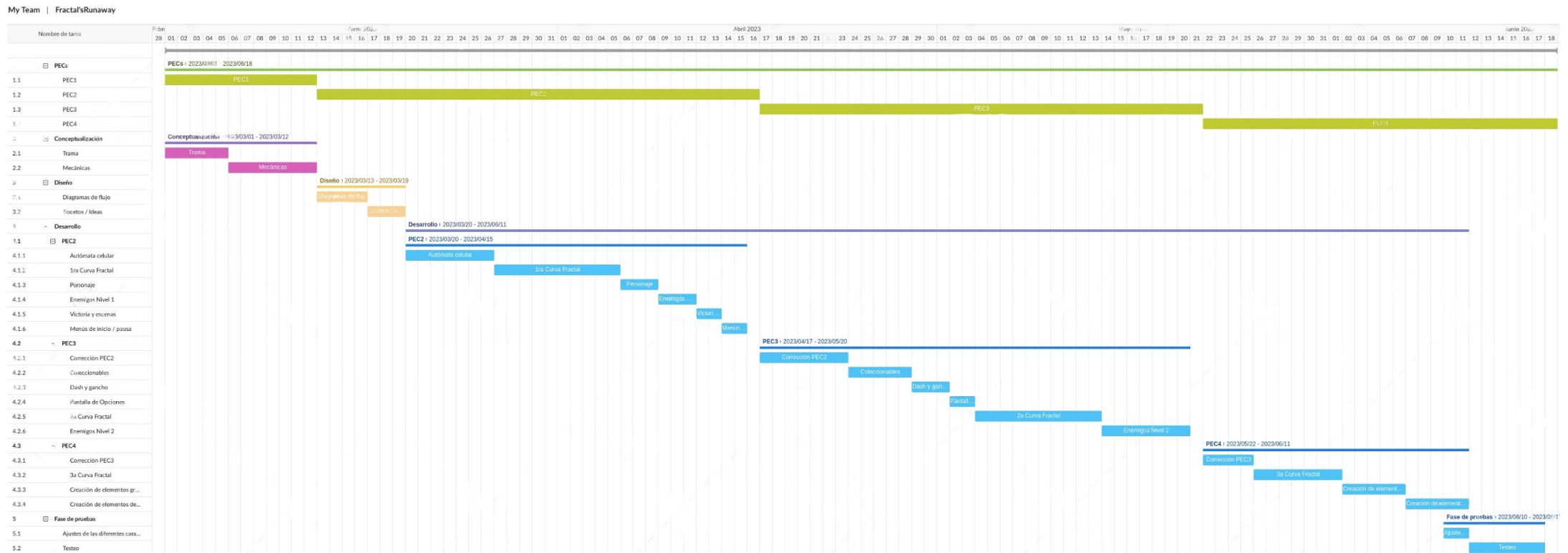
Por último, durante la fase de pruebas, se ajustarán y optimizarán los diferentes elementos del juego y se llevarán a cabo diferentes pruebas para comprobar que no existen errores antes de la entrega final.

A continuación, se adjunta el diagrama de Gantt con las diferentes fechas clave y los diagramas de flujo de los menús de navegación del juego:

En el siguiente diagrama podemos ver las diferentes fases claramente diferenciadas entre sí, siendo la de desarrollo la que más tiempo ocupa a lo largo del proyecto. A pesar de no estar indicado en el diagrama, cabe tener en cuenta que la memoria se irá completando a medida que vaya desarrollando el proyecto.

Por último, y como no estoy familiarizado con Unity ni su entorno, también se incluye un aprendizaje de estas herramientas a lo largo de todo el semestre.

Figura 1: Diagrama de Gantt (5)



A continuación, la tabla con los hitos del proyecto.

| | | Fechas | |
|-------|---------------------------------|---------------|---------------|
| | | Comienzo | Fin |
| Hitos | Entrega PEC 1 | 1 / 3 / 2023 | 12 / 3 / 2023 |
| | Implementación Autómata Celular | 20 / 3 / 2023 | 26 / 3 / 2023 |
| | Implementación 1ª Curva Fractal | 27 / 3 / 2023 | 5 / 4 / 2023 |
| | Entrega PEC 2 | 13 / 3 / 2023 | 16 / 4 / 2023 |
| | Implementación 2ª Curva Fractal | 03 / 5 / 2023 | 12 / 5 / 2023 |
| | Entrega PEC 3 | 17 / 4 / 2023 | 21 / 5 / 2023 |
| | Implementación 3ª Curva Fractal | 26 / 5 / 2023 | 01 / 6 / 2023 |
| | Entrega PEC 4 | 22 / 5 / 2023 | 18 / 6 / 2023 |

Tabla 1: Hitos

1.6. Presupuesto

Se desglosará el presupuesto en 3 partes:

Equipo Humano:

El proyecto ha sido desarrollado completamente por mí, por lo que no ha supuesto ningún coste económico. En caso de que se hubiese tenido que contratar personal para las diferentes áreas, los costes podrían haber sido los siguientes, teniendo en cuenta los diferentes perfiles profesionales:

Tabla 2: Presupuesto de equipo humano

| Puesto | Presupuesto / hora | Horas totales | Suma |
|------------------------|--------------------|---------------------|-----------------------|
| Guionista (6) | 13'5€ / hora | 5h / semana = 60h | 810€ |
| Programador (7) | 14'5€ / hora | 40h / semana = 480h | 6960€ |
| Artista 2D (8) | 15€ / hora | 30h / semana = 360h | 5400€ |
| Animador 2D (9) | 12€ / hora | 20h / semana = 240h | 2880€ |
| Técnico de sonido (10) | 10€ / hora | 20h / semana = 240h | 2400€ |
| Marketing (11) | 13€ / hora | 20h / semana = 240h | 3120€ |
| | | | Total: 21.570€ |

Los precios por hora de los diferentes puestos han sido buscados en *Glassdoor* (12), una web de búsqueda de empleo, suponiendo que el desarrollo se llevase a cabo mediante personas que cobran un sueldo en España. El sueldo en la web es anual, por lo que se ha calculado en cada caso la parte equivalente según las horas implicadas en el proyecto.

En cuanto a las horas, se han repartido de manera orientativa según la importancia en el proyecto. Por ejemplo, el programador estaría trabajando 40h semanales ya que es el que tiene más carga de trabajo. Por otro lado, el guionista trabajaría únicamente 5h a la semana ya que el guion que se pretende hacer no es demasiado elaborado ni extenso, únicamente un trasfondo.

Software:

Teniendo en cuenta que la disponemos de diferentes licencias, como las de Adobe y Unity, el coste del software también ha sido nulo. En caso de comercializar el videojuego, sí se necesitaría adquirir dichas licencias:

Tabla 3: Presupuesto de software

| Software | Precio |
|---|-----------------------|
| Pack Adobe (Illustrator, Photoshop, Adobe Animate y Adobe Audition) | 435,48€ / año |
| Unity Pro | 2040€ / año |
| Microsoft 365 | 69€ / año |
| | Total: 2544'48€ / año |

Hardware:

Se ha utilizado un PC de sobremesa, un PC portátil y 2 monitores, además de los periféricos típicos (ratón y teclado). Ya se disponía del material necesario de antemano, pero en caso de tener que comprarlo, habría sido un coste aproximado de 2500€.

1.7. Estructura del resto del documento

- Apartado 2: Análisis de mercado

En este apartado se estudia el estado del mercado actual de los videojuegos, basándome en los datos proporcionados en el Anuario 2021 por la Asociación española de Videojuegos, analizando el posible público objetivo y los diferentes perfiles de usuario.

Además, se estudia el estado del arte de los juegos de plataformas y de los algoritmos de generación procedural enfocados en videojuegos, además del uso de curvas fractales para complementarlos.

Por último, se muestra el análisis DAFO de mi proyecto.

- Apartado 3: Propuesta

En este apartado se presenta mi propuesta para el proyecto, analizando el videojuego desarrollado empleando el *framework* MDA. Además, se da una explicación del uso del autómeta celular junto a las curvas fractales, y se explica el crecimiento de dificultad a lo largo de los niveles.

- Apartado 4: Diseño

En el apartado de diseño se justifica la elección del entorno de desarrollo, los requisitos técnicos de este y las aplicaciones y herramientas utilizadas a lo largo del desarrollo.

Por otro lado, se listan los diferentes recursos gráficos y de sonido empleados para los diseños de los personajes y elementos del juego.

También, se muestran los esquemas de navegación de los menús y de dentro de un nivel.

Además, como punto a destacar, se enseñan diferentes generaciones de los tres niveles, se explica mediante imágenes el diseño de un nivel (donde aparece el jugador, los enemigos, los coleccionables y el objetivo de la meta y se contrastan los niveles generados con la propia curva del nivel, para entender el porqué de la forma.

Por último, se incluye el guion con la sinopsis del juego.

- Apartado 5: Implementación

En este apartado se listan los requisitos de instalación, así como las instrucciones de instalación y las instrucciones del propio juego.

- Apartado 6: Demostración

En este apartado se muestran los prototipos Lo-Fi y Hi-Fi del juego.

Por otro lado, se explican las pruebas realizadas a los usuarios, y se analizan los resultados de las encuestas y los cambios a los que han dado pie.

- Apartado 7: Conclusiones y líneas de futuro

En el apartado final, se analizan las lecciones aprendidas a lo largo del proyecto, se valora si se han cumplido los objetivos y planificaciones realizadas en el punto 1 de la memoria.

Por último, se mencionan los planes de futuro respecto al juego.

- Apartado 8: Bibliografía

Recoge todas las referencias empleadas a lo largo del documento.

- Anexo:

Apartado que incluye los entregables del proyecto y las encuestas realizadas.

2. Análisis de mercado

Durante las últimas décadas, el mercado de videojuegos de plataformas ha experimentado grandes cambios desde la primera aparición en las máquinas arcade (1980). Desde entonces, se han creado un sinnúmero de títulos y se han desarrollado un gran número de mecánicas que han aumentado la complejidad y jugabilidad de este género de videojuegos.

A pesar de que los juegos de plataformas se remontan a las primeras generaciones, la demanda sigue siendo fuerte gracias a franquicias como *Super Mario*, *Sonic* o *Crash Bandicoot*, que han conseguido un gran número de ventas y popularidad.

Desde la perspectiva del *framework* MDA, los tres títulos se caracterizan por sus mecánicas de movimiento (saltar, correr, deslizarse), recolección de objetos para superar los niveles y el uso de potenciadores. A nivel de dinámicas, de igual forma se centran en la habilidad y precisión del jugador para superar los diferentes obstáculos y enemigos. En cuanto a la estética, cada uno tiene la suya bien diferenciada. *Mario* es un juego relajado que transmite un sentimiento familiar, mientras que *Sonic* se diferencia por su velocidad y frenetismo y *Crash* por ser algo más cómico y caricaturesco.

Por otro lado, títulos como *Hollow Knight*, *Celeste*, *Cuphead* o *It Takes Two* han logrado destacar en el género de las plataformas gracias a una variedad de mecánicas y dinámicas que les dan un enfoque único. *Hollow Knight*, por ejemplo, destaca por su énfasis en la exploración y la lucha contra jefes desafiantes, mientras que *Celeste* se enfoca en la precisión y el dominio de las habilidades de salto. *Cuphead* combina la mecánica de plataformas con un estilo de juego de disparos de desplazamiento lateral y un estilo artístico inspirado en la animación de los años 30, mientras que *It Takes Two* se enfoca en la cooperación entre los jugadores para superar los desafíos del juego.

Dentro de los juegos de plataformas, existen diferentes subgéneros, donde cada uno destaca por una mecánica, dinámica o estética concreta:

- Plataformas básico: Es el modelo 'original' de plataformas, donde se juega en una pantalla sin desplazamientos y con diferentes plataformas y enemigos.

- Plataformas y puzles:
Son juegos en que, además de recorrer un escenario convencional de plataformas, debes realizar diferentes puzles a lo largo del recorrido.
- Plataformas con desplazamiento lateral:
Son plataformas convencionales y los más típicos, donde la pantalla se mueve de manera lateral, hasta alcanzar el final del nivel. A lo largo del recorrido, puedes encontrar diferentes elementos, como enemigos o coleccionables.
- Plataformas con desplazamiento vertical
Similar al anterior, pero en este caso, la pantalla se mueve verticalmente, hasta alcanzar la meta en la parte superior.
- Desplazamiento libre:
Conforme han avanzado los años y se han superado las limitaciones, han aparecido este tipo de plataformas, donde el escenario es bastante más grande y la pantalla puede desplazarse tanto vertical como horizontalmente.
- *Metroidvania*:
Los *metroidvania* son una variante del anterior. No obstante, mientras avanzas, vas consiguiendo habilidades o herramientas que te permiten desplazarte a zonas que anteriormente no estaban disponibles.
- *Gun&Run*:
Son juegos en los que el personaje cuenta con armas de fuego y debe ir eliminando a un gran número de enemigos según avanza en el juego.
- 3D
Siguen el mismo concepto que un juego de plataformas original, pero siendo en tres dimensiones.

Dentro de todos estos subgéneros, Fractal Runaway entraría en el de desplazamiento libre, ya que es un plataformas tradicional en el que puedes avanzar tanto vertical como horizontalmente.

2.1. Público objetivo y perfiles de usuario

Hay pocos estudios disponibles que se centren específicamente en la demografía de los jugadores de juegos de plataforma. Sin embargo, podemos hacer algunas suposiciones basadas en la demografía general de los jugadores de videojuegos. Para ello, me basaré en los datos proporcionados por la Asociación Española de Videojuegos en el Anuario 2021 (13). En este estudio se analizan los datos de 18.1 millones de jugadores.

Género:

En el anuario de 2021, se puede observar un panorama en el que, a pesar de que la proporción de jugadores de género masculino sigue siendo mayoritaria, hay un aumento en el número de jugadoras de videojuegos en comparación con años anteriores. De hecho, la brecha entre los géneros se ha reducido significativamente con el paso del tiempo.

Según los datos del anuario de 2021, el 52% de los jugadores de videojuegos son hombres, lo que equivale a aproximadamente 9.5 millones de personas, mientras que el 48% son mujeres, que suman alrededor de 8.6 millones de jugadoras.

Es cierto que, durante mucho tiempo, los videojuegos han sido considerados un territorio exclusivo para los hombres. Sin embargo, el panorama actual sugiere un cambio significativo en esta percepción. Cada vez son más las mujeres que se interesan por los videojuegos, lo que puede ser un reflejo de los cambios sociales que buscan la igualdad de género en todos los ámbitos. Es por ello por lo que se espera que esta tendencia continúe en el futuro, lo que permitirá una mayor diversidad de jugadores y una inclusión cada vez mayor de mujeres en la industria de los videojuegos.

Distribución del género en los videojuegos 2021



Figura 2: Distribución del género en los videojuegos 2021. Fuente: Anuario AEVI 2021

Edad:

Al analizar el perfil de los usuarios de videojuegos, se puede apreciar que existen diferencias significativas según la edad de los jugadores. Según el anuario de 2021, los usuarios con mayor porcentaje de consumo de videojuegos son aquellos que tienen entre 11 y 14 años, con un 78% de participación en el mercado. Este rango de edad es seguido de cerca por los jugadores de 6 a 10 años, con un 76% de participación, y por los jugadores de entre 15 y 24 años, que registran un porcentaje del 74%.

A partir de esta franja de edad, se puede observar una disminución en la participación de los usuarios en el mercado de los videojuegos. En el caso de personas de entre 25 y 34 años, el porcentaje de consumo de videojuegos disminuye ligeramente hasta el 60%. La siguiente franja de edad, comprendida entre los 35 y los 44 años, muestra un porcentaje del 39%, lo que indica una disminución significativa en el interés por los videojuegos. Finalmente, la franja de 45 a 64 años es la que presenta el menor porcentaje de consumo de videojuegos, con tan solo un 32%.

Distribución de la edad en los videojuegos 2021

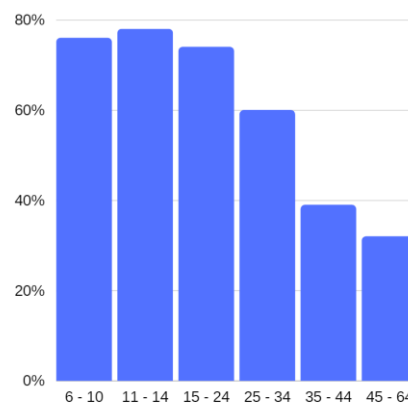


Figura 3: Distribución de la edad en los videojuegos. Fuente: Anuario AEVI 2021

Dispositivo:

Al analizar el mercado de los videojuegos, es interesante observar cómo los jugadores utilizan diferentes dispositivos para disfrutar de sus juegos favoritos. Según una gráfica reciente, se puede ver que las consolas son los dispositivos más utilizados, seguidos de los smartphones, con un 27% y un 22% de participación en el mercado, respectivamente.

Es especialmente destacable el auge del mercado de los videojuegos en los smartphones en la última década, que ha superado al mercado de PC. Esto se debe, en gran medida, a que la compra de un teléfono inteligente es más accesible y económica que la de un ordenador, lo que ha permitido que un mayor número de personas tengan acceso a los videojuegos a través de sus dispositivos móviles.

En cuanto a las consolas, además de las de sobremesa, existen también consolas portátiles que registran un 9% de participación en el mercado de los videojuegos. Asimismo, las tabletas se sitúan en un 12%, lo que indica que este dispositivo también se utiliza para jugar.

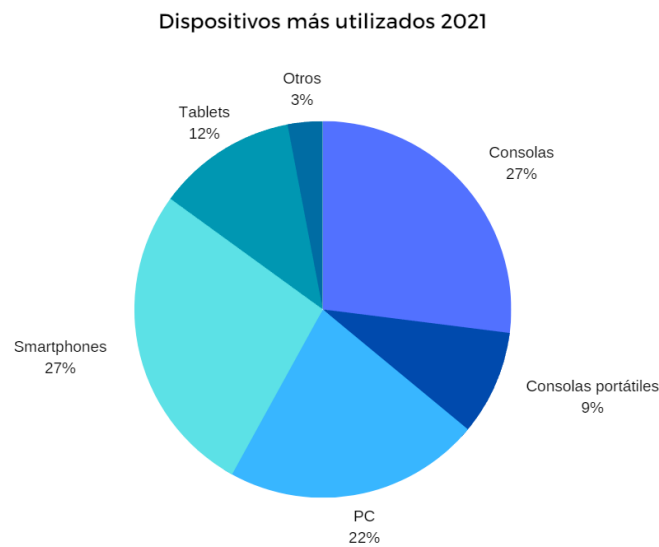


Figura 4: Dispositivos más utilizados. Fuente: Anuario AEVI 2021

Frecuencia de uso:

Dentro del ámbito europeo, España ocupa el quinto lugar en cuanto a horas dedicadas a los videojuegos, con una media de 8.1 horas a la semana.

Este dato nos sitúa por debajo de países como Reino Unido, con 10.6 horas, Alemania con 9.2 horas, Italia con 8.6 horas y Francia, que nos iguala en 8.1 horas. Este ranking nos permite comprender el importante papel que los videojuegos juegan en la cultura y el entretenimiento de la sociedad europea.

Si profundizamos en el análisis de las horas de juego en España, se puede observar que el 56% de las horas son jugadas por hombres y el 44% por mujeres, lo que indica una

tendencia a la equiparación en la participación de ambos géneros en el mundo de los videojuegos.

Además, es interesante conocer el desglose de las horas de juego por dispositivo. Según el estudio, las consolas son el dispositivo que ocupa el primer lugar en cuanto a horas de juego, con 5.5 horas a la semana (67% hombres, 33% mujeres), seguidas por los smartphones y tabletas, con 5.1 horas a la semana (44% hombres y 56% mujeres) y el PC, con 4.1 horas a la semana (59% hombres, 41% mujeres).

Géneros más populares:

Dentro del mercado de videojuegos, podemos encontrar diferentes géneros que compiten por la atención y preferencias de los jugadores. Según las ventas, los videojuegos de acción lideran el ranking de popularidad con 2.348.882 millones de ventas, seguidos de los juegos de deporte con poco más de 1 millón de ventas, los juegos de rol con casi 900.000 ventas y los de aventuras con algo más de 850.000.

Es importante destacar que los juegos de plataformas pueden incluirse en algunos de estos géneros, ya que su jugabilidad y mecánicas pueden variar según el enfoque del desarrollador.

2.2. Estado del arte

Tras analizar los datos recolectados en los apartados previos, he decidido comenzar con el desarrollo de un juego de plataformas. Aunque no es el género más popular, creo que su encanto y diversión son atemporales.

Creo que el género de plataformas puede resultar atractivo para cualquier público debido a la variedad de mecánicas y los desafíos que se presentan, pudiendo ser disfrutado tanto por jugadores más casuales como por aquellos que buscan un reto más complejo.

Además, los juegos de plataformas son fáciles de adaptar a diferentes plataformas. Sus controles son relativamente sencillos en la mayoría de los casos, lo que permite disfrutar de la experiencia de juego tanto en el teclado y ratón, como en el mando, e incluso en la pantalla táctil. Esto significa que el juego puede llegar a una amplia audiencia, independientemente del dispositivo que se esté utilizando.

Como el punto principal de mi proyecto es el uso de algoritmos de generación procedural como solución al problema de la generación de diferentes recursos en los videojuegos, a continuación, daré un poco de contexto mencionando algunos títulos que también dan uso a esta tecnología.

Durante los últimos años, la generación de contenido de manera procedural ha sido un campo de gran importancia, tanto a nivel teórico como práctico en la industria, ya que permite ahorrar recursos en el desarrollo de videojuegos. Los algoritmos de generación procedural pueden aplicarse a muchos niveles: música, diseño de entornos, sistemas, comportamiento de NPC y un largo etcétera.

The Binding of Isaac y *Enter the Gungeon* son dos ejemplos de juegos que implementan algoritmos de generación procedural en su diseño. Estos juegos han optado por la reusabilidad del escenario, permitiendo al jugador enfrentar los mismos retos, pero con entornos distintos en cada partida.

En The Binding of Isaac, los escenarios se generan en una red de cierto tamaño, donde se cargan un número de salas ya prediseñadas. Dependiendo del punto de partida, se añaden más o menos salas al nivel. Además de las salas comunes, el juego siempre genera un número de salas especiales, como la sala de inicio, la del enemigo final del nivel o una tienda.

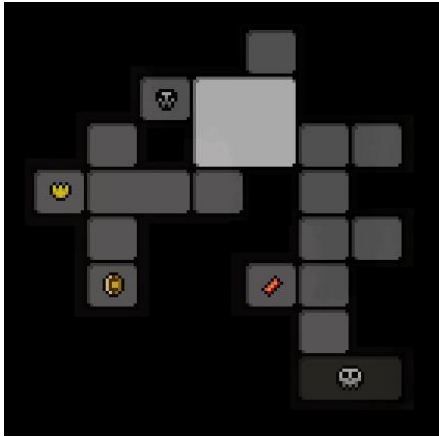


Figura 5: Ejemplo 1 mapa The Binding of Isaac. (42)



Figura 6: Ejemplo 2 mapa The Binding of Isaac. (41)

Por su parte, Enter The Gungeon también sigue una fórmula similar a la anterior, pero utilizando patrones para generar las diferentes salas. En ambos juegos, la generación de contenido de manera procedural permite que cada partida sea única y que el jugador experimente diferentes desafíos.



Figura 8: Ejemplo 1 mapa Enter the Gungeon. (43)

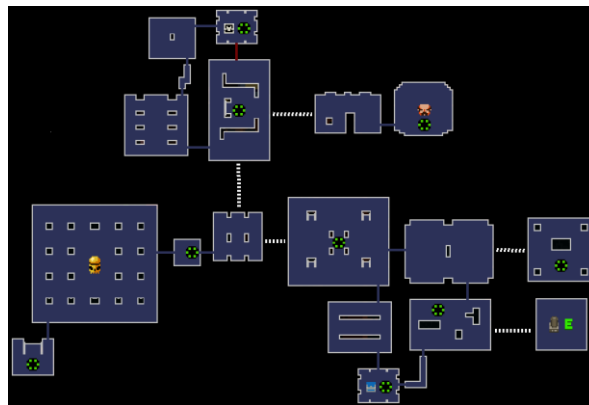


Figura 7: Ejemplo 2 mapa Enter the Gungeon. (43)

Podemos analizar cómo estos juegos utilizan la generación de mundo para lograr una experiencia jugable interesante mediante el framework MDA. A nivel de mecánicas, la generación procedural permite que los jugadores enfrenten desafíos diferentes en cada partida, añadiendo variedad y reusabilidad. En el nivel de estética, la generación de mundo permite que cada partida tenga su propia identidad visual y sorprenda al jugador con entornos nuevos e inesperados. Por último, en el nivel de historia, la generación de mundo permite que cada partida tenga una narrativa diferente, aunque se mantengan los mismos objetivos y desafíos en cada una de ellas.

Para el desarrollo de mi proyecto, como método de generación procedural, he decidido implementar un autómata celular, acompañado de una curva fractal.

Un autómata celular (3) es un modelo matemático que se utiliza para la simulación de sistemas complejos a través de la división del espacio y el tiempo en celdas discretas. Cada celda puede estar en uno de los estados posibles y siguen un conjunto de reglas para estar en un estado u otro, basándose en las celdas vecinas. Para el análisis de las celdas vecinas, existen dos tipos de vecindades muy utilizadas, que son la de Von Neumann (14) y la de Moore (15), donde cada una de ellas emplea un rango y una fórmula para saber el número de vecinos para tener en cuenta.

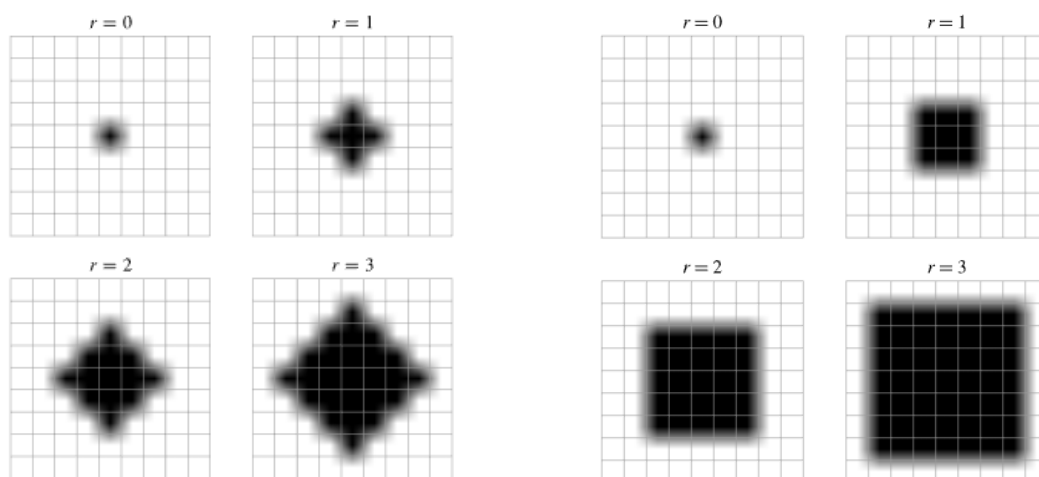


Figura 9: Vecindario de Von Neumann (izquierda) y vecindario de Moore (derecha). (16)

La vecindad de Von Neumann con rango 1 es una disposición en forma de cruz, en la que una celda puede tener cuatro celdas vecinas, arriba, abajo y a los lados. Por tanto, para tener en cuenta el estado de la celda central, se tienen en cuenta las 4 celdas que la rodean.

Por otro lado, la vecindad de Moore con rango 1 es una disposición cuadrada, donde una celda tiene ocho celdas vecinas, formando un cuadrado a su alrededor. Por tanto, para tener en cuenta el estado de una celda, esta tendrá en cuenta a las 8 celdas que la rodean.

Para cualquiera de las dos vecindades, una celda puede cambiar en función de las reglas establecidas, que definirán como una celda interactúa con sus vecinas.

En mi caso, utilizaré un autómata celular empleando la vecindad de Moore para generar el escenario jugable. La regla que determina si una celda cambia de estado será comprobando los vecinos. En caso de que más de cuatro tengan valor 1, la celda pasará a tener valor 1. Por otro lado, en caso de que más de cuatro tengan valor 0, la celda pasará a tener valor 0.

De este modo, las celdas de mi autómata tendrán dos posibles estados: fondo (valor 0) o plataforma (valor 1). Las plataformas serán celdas con colisión, donde el jugador podrá saltar, caminar y correr. Por otro lado, el fondo no tendrá colisión, de modo que será el espacio donde el jugador podrá moverse.

A dicho autómata, le 'acompañará' una curva fractal. Una curva fractal es una curva que presenta autosimilitud a diferentes escalas. Esto quiere decir que, al observarla a diferentes niveles de detalle, se observan patrones diferentes que se repiten a lo largo de la curva. Los fractales se utilizan en diversas áreas de la ciencia y la tecnología para moldear fenómenos naturales, como podría ser la formación de montañas, la geometría de las hojas o la estructura de los copos de nieve. Existen diversas curvas fractales, como la de Hilbert (17), la de Koch (18) o la de Peano (19), entre otras.

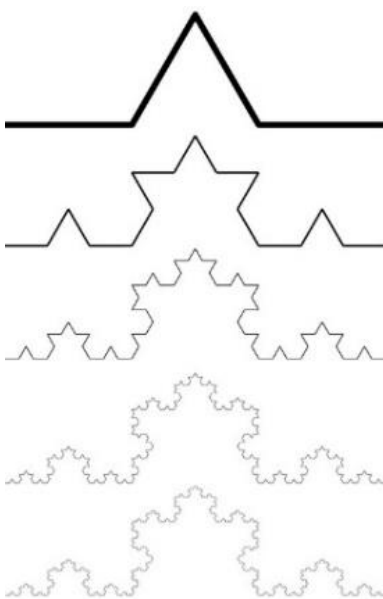


Figura 11: Primeros 5 pasos de la curva fractal de Koch. (20)

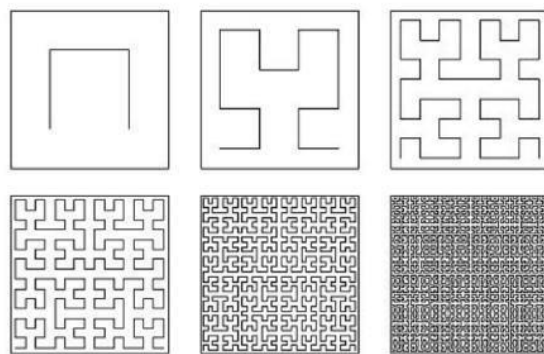


Figura 10: Primeros 3 pasos de la curva fractal de Hilbert. (44)

Al implementar la curva fractal dentro del autómata celular, la generación del espacio seguirá el patrón realizado por la curva. De este modo, se seleccionan los diferentes puntos que forman la curva y se marca cada celda que se encuentra en su vecindad con el valor deseado dentro de los estados disponibles, que en mi caso es el estado 0. Por tanto, el autómata irá generando el escenario teniendo como guía la curva, de modo que se obtendrá una estructura mucho más compleja y detallada.

He decidido utilizar un autómata junto a una curva fractal ya que permite una aleatoriedad controlada. Mediante un autómata, puedo controlar el grado de aleatoriedad a la hora de generar los patrones y estructuras, permitiendo así generar experiencias de juego coherentes y predecibles. Además, a nivel computacional requieren una cantidad relativamente baja de cálculos y recursos, lo cual permite crear escenarios de un tamaño considerable sin demasiado problema. Además, añadiéndole la curva, puedo controlar hasta cierto punto que el espacio generado siga un patrón, permitiéndome así obtener un escenario con sentido y diferentes caminos a seguir, de modo que llegar hasta el punto final no sea simplemente una línea recta y presente un mayor reto para el jugador.

Hay algunos videojuegos que utilizan la implementación de un autómata junto a una curva fractal, como podría ser el caso de *Galak-Z* (21). *Galak-Z* es un juego de acción de desplazamiento lateral en el que los jugadores pilotan una nave espacial y luchan contra enemigos en diferentes escenarios. Las mecánicas del juego incluyen la capacidad de disparar y maniobrar la nave, recoger objetos y habilidades especiales. Pero, lo relevante en relación con mi proyecto, es la generación de estructuras mediante un autómata celular y una curva fractal, más concretamente la de Hilbert (17). Mediante esto, el juego consigue ahorrar una gran cantidad de trabajo de diseño de escenarios, pero manteniendo un mundo vivo, complejo y variado.

La idea para implementar un autómata junto a una curva fractal me vino dada por el artículo '*Improving procedural 2D map Generation based on multi-layered cellular automata and Hilbert curves*', donde se pueden ver analizadas este tipo de implementaciones más detalladamente, ya que se pone en práctica el uso de estas dos tecnologías y se ven los diferentes resultados que se pueden obtener (22). Asimismo, para la implementación del autómata celular me basé en el código del blog '*PavCreations*', que seguía las bases del

artículo anteriormente mencionado, donde generaban un escenario aleatorio mediante un autómatata y le aplicaban la curva de Hilbert (23).

Para mi proyecto, me pareció interesante aplicar diferentes curvas fractales, de modo que se pueda ver las características de cada una y los diferentes niveles que generan. Concretamente, utilizaré tres, una para cada nivel.

Para la construcción de dos de las curvas, la de Moore y la de Peano, he empleado el sistema de Lindenmayer o sistema-L (24), un modelo matemático empleado para generar fractales, además de para modelar el proceso de crecimiento de plantas y la morfología de varios organismos. En un sistema-L se utilizan un conjunto de símbolos llamado alfabeto que representan a los elementos de la estructura que se modela. Este, define los componentes básicos que se utilizarán en las reglas de producción, las cuales, a su vez, especifican como se deben expandir los símbolos en cada iteración. Por último, un axioma es la cadena inicial de símbolos a partir de la cual se inicia la generación del sistema.

En el primer nivel, aplicaré la curva de Moore (25), que es una variante de la curva de Hilbert. Según el sistema de Lindenmayer, se expresa de la siguiente manera:

Alfabeto: L, R
Constantes: F, +, -
Axioma: LFL + F + LFL
Reglas de producción:
 L → - RF + LFL + FR -
 R → + LF - RFR - FL +

Donde F significa 'avanzar', - significa 'girar a la izquierda 90°' y + significa 'girar a la derecha 90°'.

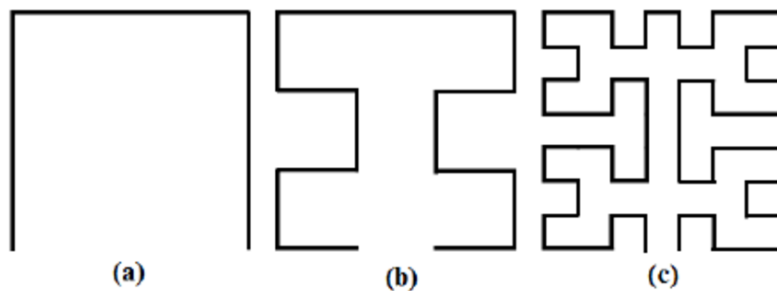


Figura 12: Primeros 3 pasos de la curva de Moore. (26)

En el segundo nivel, aplicaré la curva de Peano (19), al igual que la de Moore, esta también puede ser generada mediante el sistema- L de la siguiente manera:

Alfabeto: L, R

Constantes: F, +, -

Axioma: L

Reglas de producción:

L → LFRFL-F-RFLFR+F+LFRFL

R → RFLFR+F+LFRFL-F-RFLFR

Donde F significa 'avanzar', - significa 'girar a la izquierda 90°' y + significa 'girar a la derecha 90°'.

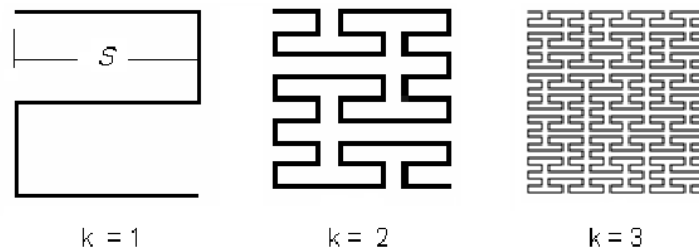


Figura 13: Primeros 6 pasos de la curva de Peano. (27)

Por último, en el tercer nivel aplicaré la curva de Sierpinski (28). En este caso, para generar la curva se ha empleado un algoritmo recursivo que la genera.

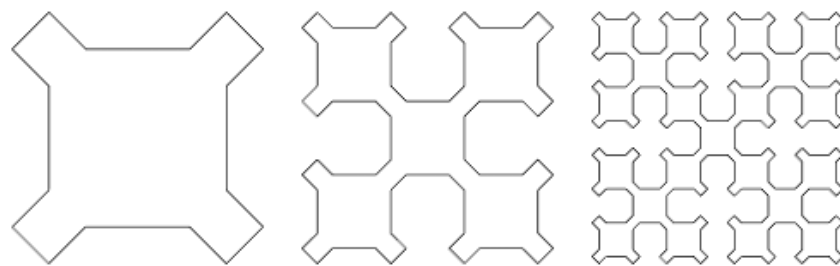


Figura 14: Primeros 3 pasos de la curva de Sierpinski. (16)

Mediante estas tres curvas, el proyecto aporta conocimiento sobre el impacto que tiene la implementación de diferentes tipos de curvas fractales, así como la ya mencionada reusabilidad al juego y, por consiguiente, la reducción los costes de producción a la hora de crear nuevo contenido.

2.3. Análisis DAFO

ANÁLISIS DAFO



Figura 15: Análisis DAFO. (29)

3.Propuesta

3.1. Definición de objetivos/especificaciones del producto

El proyecto propuesto se enmarca en el *framework* MDA y consiste en un videojuego de plataformas 2D desarrollado en Unity, que cuenta con diferentes niveles aleatorios que el jugador deberá superar. El objetivo principal del juego es avanzar por el escenario hasta alcanzar el punto final, evitando a los enemigos y tratando de recolectar elementos que conformarán una 'llave' que abrirá la salida al siguiente nivel.

A nivel de mecánicas, no he querido hacer algo demasiado complicado, por lo tanto, en cuanto a controles me he centrado en: desplazamiento lateral, salto y escalada. Además, he añadido un gancho y un deslizamiento para hacer el juego más ágil. Con tal de concentrar mis esfuerzos entorno a la generación de mundo, los enemigos no tienen una lógica demasiado compleja. En el primer nivel, se desplazarán lateralmente hasta caerse por un precipicio, o hasta chocarse con una pared, momento en que se darán la vuelta. En el segundo nivel, perseguirán al jugador hasta que salga de su radio de acción o se choquen con una pared, y el tercer nivel será una combinación de los dos anteriores.

En cuanto a las dinámicas, el juego se centra en la movilidad, ya que permite decidir al jugador como jugar: de manera rápida y agresiva, moviéndose rápidamente alrededor del escenario; o de manera tranquila y estratégica, intentando no llamar la atención de los enemigos. La generación procedural es interesante en este sentido, ya que, al no haber nunca dos niveles iguales, se podrá superar cada uno de diferente manera.

Por último, en cuanto a la estética, decidí darle un toque futurista ya que es algo que, a pesar de que se utiliza en muchos videojuegos, me atrae desde siempre. Además, teniendo en cuenta la música y paleta de colores empleada, también tendrá un ambiente con un toque tenso que puede generar situaciones divertidas.

En cuanto a la integración del autómatas celular en el juego, se utilizará para la generación de los diferentes elementos del escenario, como las plataformas y la ubicación de los enemigos. Como ya he mencionado en el punto anterior, cada uno de los tres niveles aplicará una curva fractal diferente, con tal de añadir incluso más variedad a la ya dada por el autómatas.

Al generarse, cada nivel se tendrá en cuenta diferentes valores, como podrían ser el porcentaje de escenario que es fondo y el porcentaje que es plataforma, o el grosor de la curva, que hará que el recorrido sea más o menos estrecho.

Teniendo en cuenta que cada una de las curvas implementadas ocupa un espacio diferente, he decidido que todas sean de orden dos, pero con diferentes longitudes de línea. Así, el nivel uno que implementa la curva de Moore con longitud de línea 12, generando un mapa bastante pequeño, debido a la sencillez de la curva. A continuación, en el nivel dos, la curva de Peano, con longitud 8, generará un mapa de tamaño medio, ya que se trata de una curva algo más compleja. Por último, en el nivel tres, la curva de Sierpinski, con longitud 7, generará un mapa bastante grande, debido a que se trata de una curva que ocupa un gran espacio.

Se han escogido estos valores para el orden de las curvas, ya que una curva de orden 1 genera un espacio demasiado pequeño y una curva de más de orden 3 genera un espacio demasiado grande, pudiendo complicar la generación del nivel. Las longitudes de línea se han escogido tras diferentes pruebas, de modo que el tamaño del mapa crezca a medida que se avanza en el juego, sin ser excesivo.

Por tanto, la dificultad radicará en tres factores: el primer lugar, los enemigos. Cada nivel tendrá enemigos con un comportamiento, siendo el nivel 1 el más sencillo, ya que los enemigos serán fácilmente esquivables, porque su patrón de movimiento se basa en el movimiento lateral hasta chocarse con una pared, momento en que darán la vuelta, o caerse por un precipicio. En el nivel 2, la dificultad aumenta ligeramente, ya que los enemigos, una vez entres en un su radio, te perseguirán hasta que salgas del mismo o se choquen con alguna pared. Por último, el nivel 3 contendrá los dos tipos de enemigos anteriores, siendo más complicado moverse por el mapa.

El segundo factor que hará que el juego aumente la dificultad según se llega más lejos son los escenarios. Como he comentado anteriormente, cada nivel aplica una curva diferente con longitud de línea también diferente. De este modo, el nivel 1 hay un mapa de 55x55, de 85x85 en el nivel 2 y de 110x110 en el nivel 3.

El tercer factor que hará que el juego sea más complicado según avanzan los niveles es la propia generación procedural, ya que en cada ejecución se presentará un escenario completamente diferente a los anteriores. Además, en cada nivel se deberán encontrar las diferentes llaves que permiten completar el nivel. El número de llaves variará en función del nivel, siendo el mínimo una y el máximo tres.

Por otro lado, el juego cuenta con diversos paneles de menú, que permiten navegar entre ellos o aparecen tras completar / ser derrotado en un nivel. Mediante estos, el jugador podrá leer más acerca de la sinopsis del juego, sobre los controles o los créditos. Además, se implementa un menú de opciones, que permite la modificación de ciertos valores, como podrían ser la resolución, los gráficos o el volumen.

Por último, el juego implementa diseños gráficos creados exclusivamente para el desarrollo de este proyecto. Concretamente, se han añadido recursos gráficos, como el propio personaje principal, los enemigos, los coleccionables a recoger y todos los escenarios. Por otro lado, a nivel de sonido, se implementan dos pistas de audio, una para el menú y otra para los niveles, además de diversos efectos de sonido.

3.2. Modelo de negocio / Estrategia de marketing

El proyecto desarrollado no tendrá una puesta en marcha, pues no es mi objetivo que se comercialice. Este será utilizado únicamente como entrenamiento práctico en el desarrollo de videojuegos, ámbito al que me quiero dedicar en un futuro. Además, podría emplearse a modo de experimento para estudiar el impacto que las diferentes curvas fractales ejercen sobre la manera de jugar de los usuarios.

4. Diseño

4.1. Entorno

Al comenzar el proyecto, estuve debatiéndome por Unity o Godot. En el caso de Godot, es una herramienta más reciente que está ganando relevancia en el mundo del desarrollo de videojuegos, y que permite programar en GDScript (su propio lenguaje), Visual Scripting, C# y C++, mientras que Unity permite programar en C# y lleva más años funcionando, siendo de los motores más relevantes hoy día.

Finalmente me acabé decantando por Unity, ya cuenta con muchos tutoriales, documentación, blogs de desarrollo y repositorios con código que hacen más fácil el aprendizaje. Si bien es cierto que recientemente Godot está en auge, no consigue alcanzar a Unity en este aspecto.

Además, hace unos años tuve la oportunidad de trabajar con Unity en mis estudios previos, por lo que me ha resultado sencillo adaptarme a su funcionamiento. También, gracias a cursos internos dentro de mi empresa, pude realizar un pequeño curso de Udemy (30), de manera gratuita, que me permitió reaprender el funcionamiento básico de la herramienta y me proporcionó los elementos gráficos que he en las entregas previas del juego.

4.2. Requisitos técnicos del entorno

Los requisitos de Unity para la versión en la que he trabajado (2021.3.19f1) para Windows son los siguientes:

Tabla 4: Requisitos de Unity 2021

| | Versión del sistema operativo | Graphics API |
|---------|--|--|
| Windows | Windows 7 SP1 o posterior, Windows 8, Windows 10 y Windows 11, solo versiones de 64-bit. | GPU compatibles con DX10, DX11, y DX12 |
| macOS | Mac OS 10.13 o posterior | GPU Intel y AMD compatibles con Metal |

En mi caso, el equipo empleado para el desarrollo del proyecto con Unity 2021 supera los requerimientos:

Tabla 5: Características del ordenador empleado

| | Versión del sistema operativo | Graphics API |
|---------|-------------------------------|--|
| Windows | Windows 11 64-bit. | GPU compatibles con DX12 (Nvidia GeForce rtx 3060) |

4.3. Herramientas utilizadas

A lo largo del desarrollo, se han empleado las siguientes herramientas:

- Unity 2021.3.19f1: Herramienta elegida para el desarrollo del proyecto, mediante la cual se crea el videojuego y se exporta. (2)
- Word: Editor de texto de Microsoft que permite la creación del guion y la redacción de la memoria. (31)
- GitLab: Es una herramienta de código abierto que permite almacenar código y tener un control de versiones de este. (32)
- GitLab Bash: Herramienta que se instala en el equipo para poder sincronizar el proyecto en local con el repositorio de GitLab.
- GanttPro: Herramienta que permite crear un diagrama de Gantt. (5)
- Streamlabs OBS: Software gratuito que permite grabar videos y realizar streaming. Permite crear una escena con diferentes elementos, así como la cámara, la pantalla o aplicaciones concretas. Utilizado para grabar los videos de las diferentes entregas. (33)
- Notion: Herramienta de gestión de proyectos y toma de notas, utilizado para gestionar el desarrollo e ir marcando objetivos. (34)
- Canva: Software web empleado para los diferentes gráficos y diagramas. (29)
- Pixilart: Software web para el diseño píxel art de los escenarios. (35)
- PowerPoint: Software de Microsoft empleado para el prototipo Hi-Fi. (31)
- Creately: Herramienta online de creación de diagramas, empleada para los prototipos Lo-Fi y Hi-Fi. (36)
- Google Forms: Herramienta de Google que permite crear formularios online, empleada para las pruebas con usuarios. (37)

4.4. Assets y recursos

Los diferentes elementos gráficos del personaje, enemigos del primer nivel, llaves y salida han sido creados por Alex Ventura (38), que se prestó voluntario para hacer los diseños. Los escenarios y los enemigos del segundo nivel han sido diseñados por mi mediante la herramienta Pixilart (35).

A continuación, mostraré los diferentes recursos para los personajes y escenas:

- Personaje protagonista:

Es el personaje controlado por el jugador. Cuenta con diferentes animaciones, así como correr, saltar, escalar, hacer un *dash* y la animación de muerte.

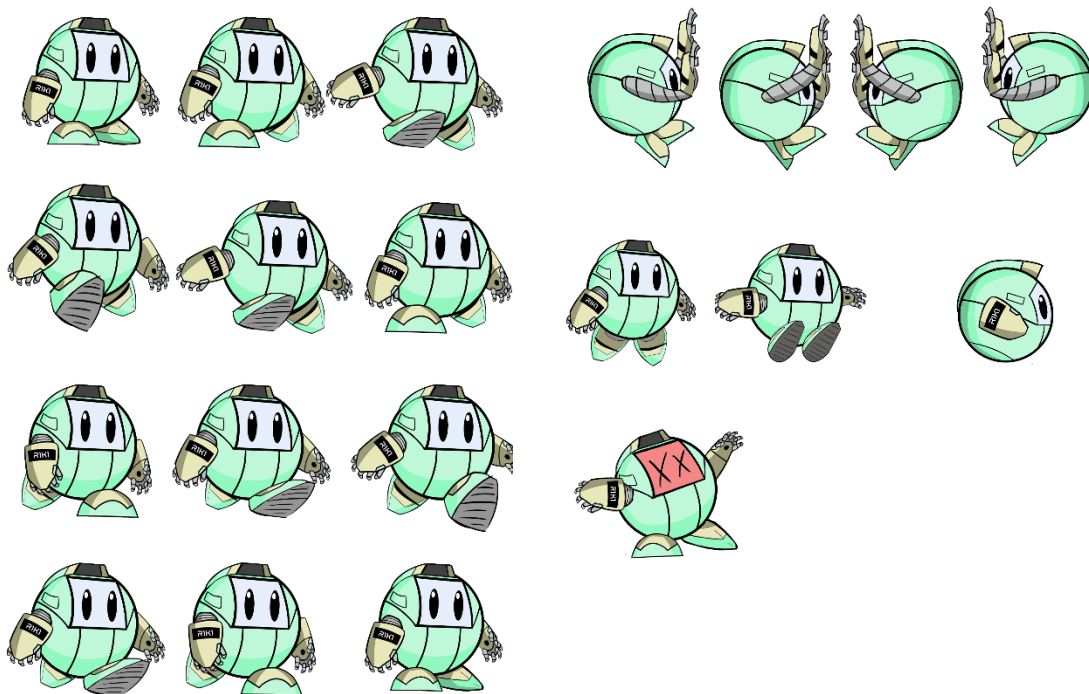


Figura 16 : Sprites del protagonista

- **Enemigos:**
Personajes que presentan un desafío al protagonista, habiendo de esquivarlos para no perder puntos de salud.



Figura 17: Enemigos nivel 1.



Figura 18: Enemigos nivel 2.

- **Coleccionables:**
Diseño de los diferentes elementos coleccionables que necesita el protagonista para pasar al siguiente nivel.

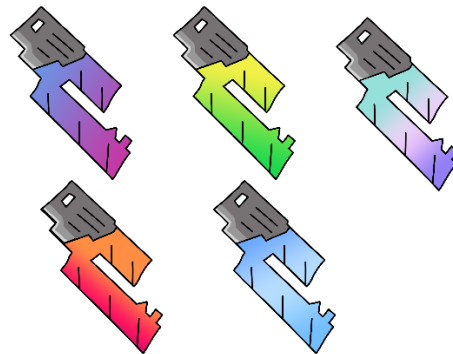


Figura 19: Diseños de las llaves.

- **Puntos de salud:**
Diseño de los puntos de salud del protagonista:



Figura 20: Sprite de un punto de salud.

- Tiles de entorno:

Diseño del escenario, que crea el autómata celular al ejecutarse. Existen dos tipos, fondo y plataforma.

- Fondo:

Diseño de los diferentes tiles de fondo:



Figura 21: Tile de fondo.

- Plataforma:

Diseño de los diferentes tiles de plataformas:

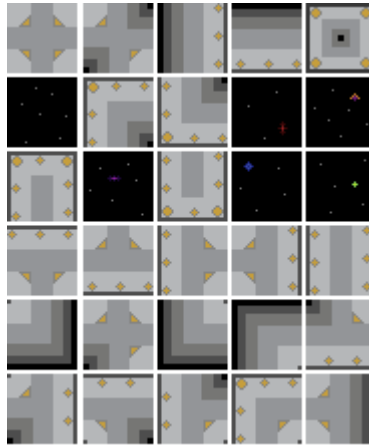


Figura 22: Tiles de plataformas.

- Salida:

Sprite de la 'salida' que hace las veces de fin del nivel.

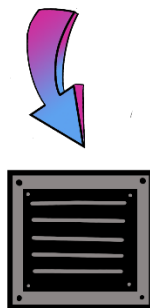


Figura 23: Sprite de salida del nivel.

En el desarrollo del juego, también se han incluido recursos de sonido, como podrían ser pistas de audio o clips. Estos han sido producidos por Leonardo Meretta, que se presentó voluntario para echarme una mano.

Concretamente, se cuentan con 2 pistas de audio, una para los menús y otra para los niveles.

Además, se han creado también diferentes clips para los efectos de sonido, tanto del personaje como del entorno.

4.5. Esquemas de arquitectura

A continuación, se adjuntan los diagramas de navegación entre menús y dentro de un nivel. En el primer caso, podemos ver las diferentes pantallas (en azul) y botones de cada una (en negro) que aparecerán a lo largo del juego.

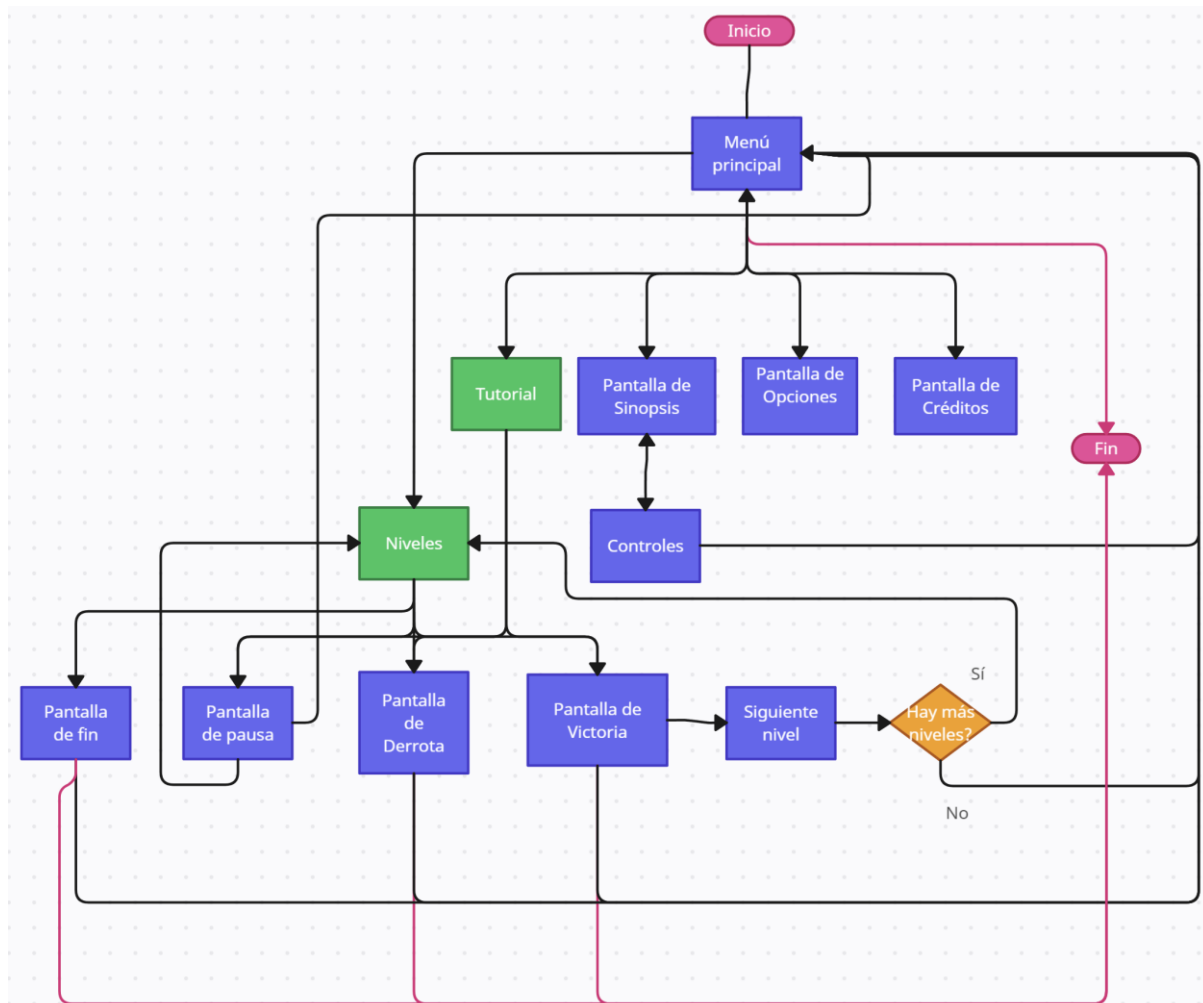


Figura 24: Esquema de los menús del juego. (36)

Como vemos, al iniciar el juego, la primera pantalla es la de nivel principal. Una vez allí, tenemos opción de consultar créditos, controles y opciones, que son pantallas con diferentes funciones que nos permitirán volver de nuevo al menú. Además, tenemos la opción de salir del juego o de jugar un nivel. Una vez le damos a jugar, en función de lo que suceda durante el nivel, se nos mostrará una pantalla u otra.

En el siguiente diagrama, se ven las diferentes posibilidades a seguir dentro de un nivel. Como vemos, en caso de que el jugador se quede sin vida, finalizará el nivel mostrando la pantalla de derrota. Cuando llega a la meta, se comprobará que haya obtenido todos los coleccionables necesarios, en caso de que no continuará el nivel y se mostrará por pantalla un mensaje. En caso de que tenga todos los coleccionables, y si hay un siguiente nivel, pasará a este, sino volverá al menú principal.

4.6. Diseño de niveles

Fractal Runaway cuenta con 3 niveles, donde en cada uno se aplica una curva fractal diferente. En cada uno de los niveles, el objetivo principal llegar a la salida sin haber perdido los puntos de salud, y habiendo recogido las diferentes llaves que hay alrededor del mapa.

Con tal de desplazarse a lo largo del nivel, el jugador cuenta con diferentes opciones de movimiento. En primer lugar, podrá moverse horizontalmente con las teclas A y D. También cuenta con la capacidad de saltar y realizar un segundo salto en el aire. Debido a que la generación es aleatoria y no se sabe exactamente como es el mapa generado, decidí darle otras opciones para asegurar de que pueda superar muros altos, como la capacidad de escalar, o el uso de un gancho que llevará al jugador hasta el lugar al que apunte al ratón (siempre y cuando no quede fuera del rango de acción). También se cuenta con un impulso para moverse más rápidamente.

En el juego, se cuenta también con la presencia de entidades enemigas, que supondrán un obstáculo para el jugador, haciéndole perder un punto de salud en caso de entrar en

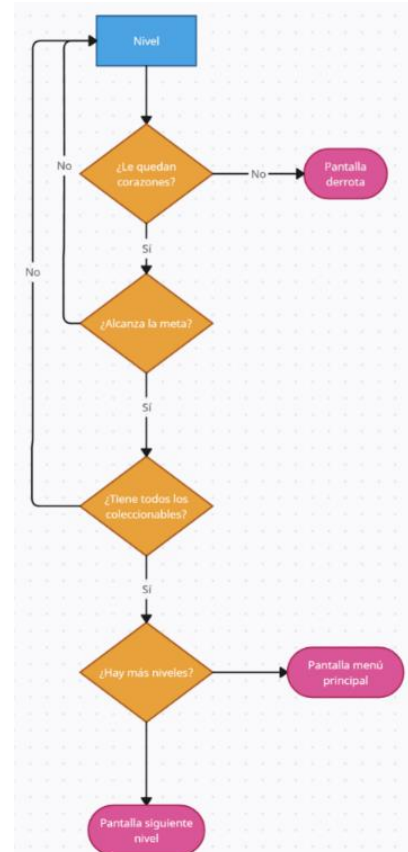


Figura 25: Esquema de navegación dentro de un nivel. (36)

contacto con alguno de ellos. En el nivel 1, los enemigos se moverán por la zona donde hayan aparecido, dando media vuelta al chocar con una pared o cayendo al vacío en los precipicios. En el nivel 2, flotarán y seguirán al jugador una vez este entre en su radio, hasta que se salga del mismo o que choque con una pared. En el nivel 3, se mezclarán los dos tipos de enemigos de los niveles anteriores, siendo así más complicado para el jugador.

A continuación, se adjuntan varias capturas de las diferentes generaciones del primer nivel:

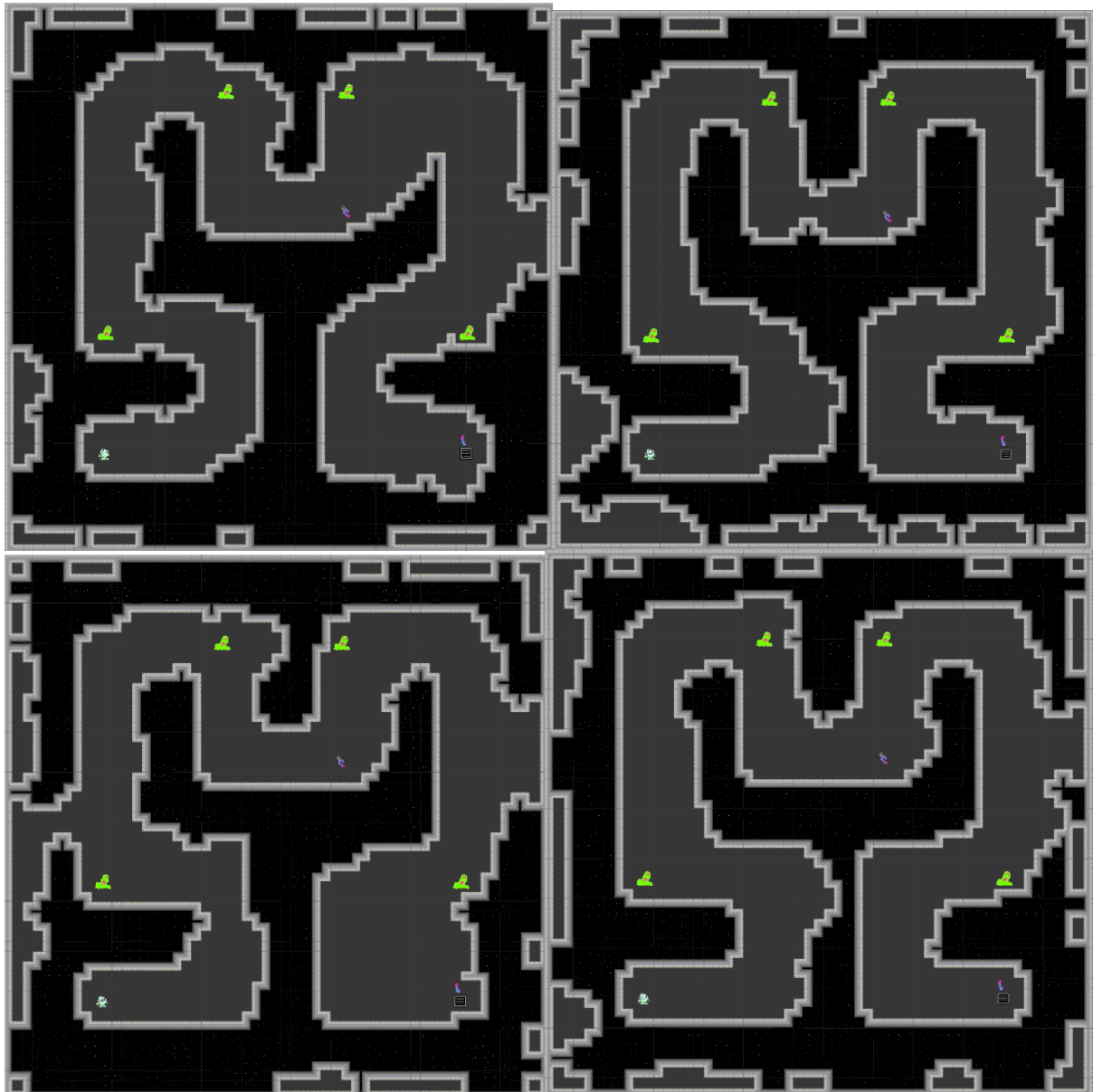


Figura 26: 4 generaciones del nivel 1.

Veamos ahora ejemplos para el segundo nivel, donde se aplica la curva fractal de Peano, se generarán diez enemigos y dos coleccionables:

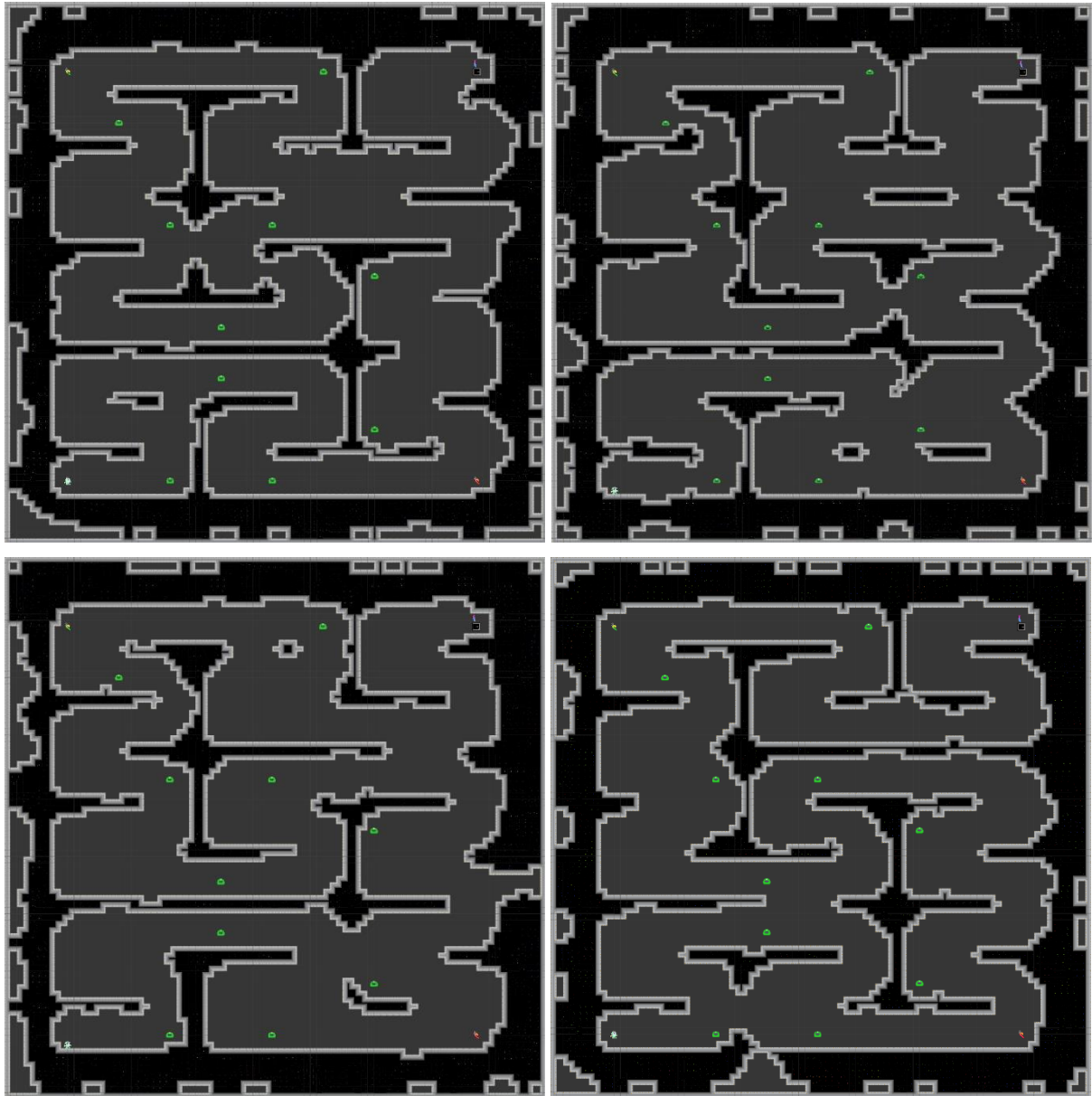


Figura 27: 4 generaciones del nivel 2.

Veamos ahora ejemplos para el tercer nivel, donde se aplica la curva fractal de Sierpinski, se generarán un número diferente de enemigos y coleccionables, para adaptarlos al espacio de la curva:

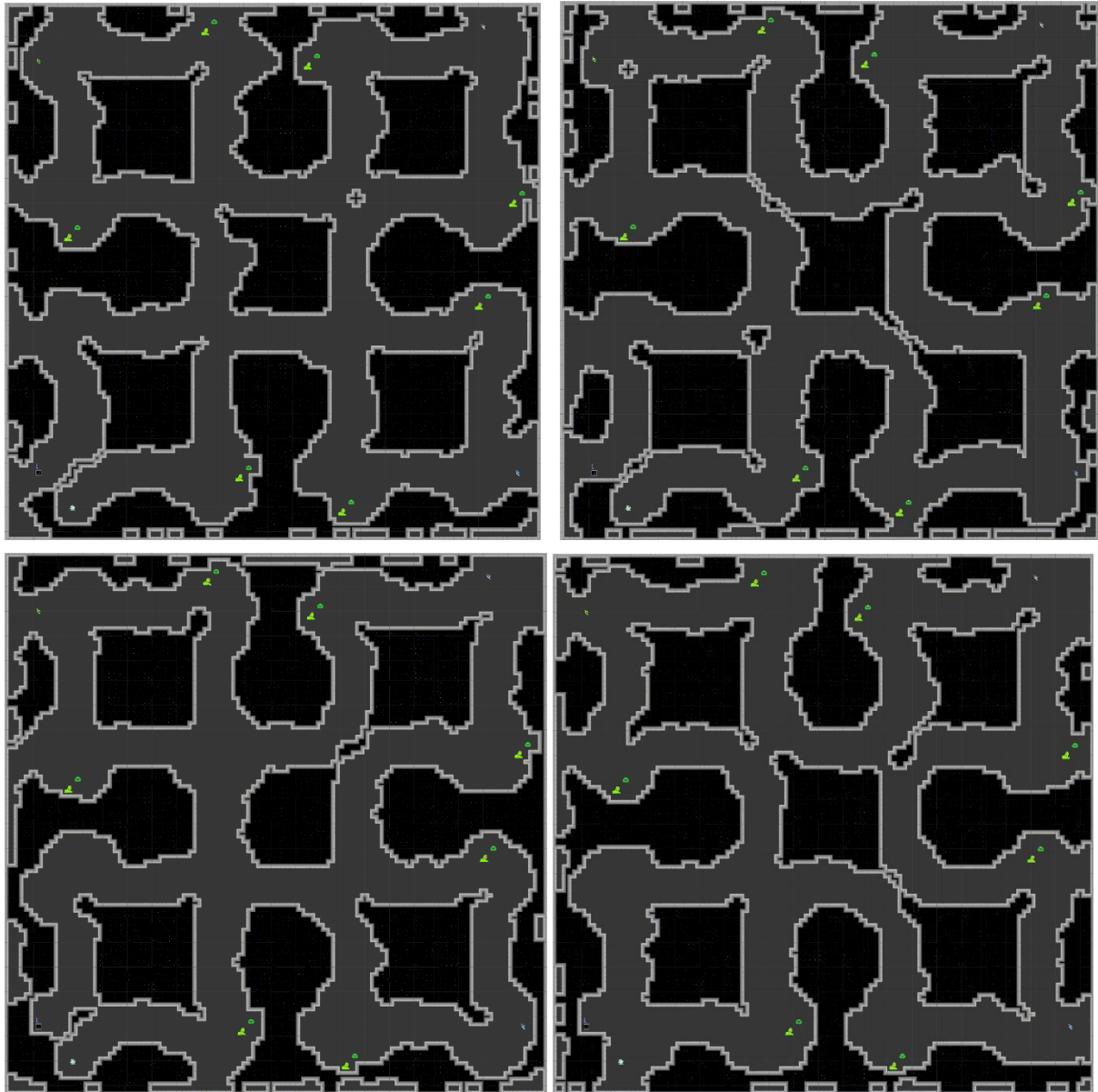


Figura 28: 4 Generaciones nivel 3.

Como podemos ver, cada generación es ligeramente diferente entre sí, variando entre niveles y dando así variación al juego. Los ejemplos anteriores siguen la forma de la curva de Moore de orden 2, que es la curva fractal que se aplica en el primer nivel:

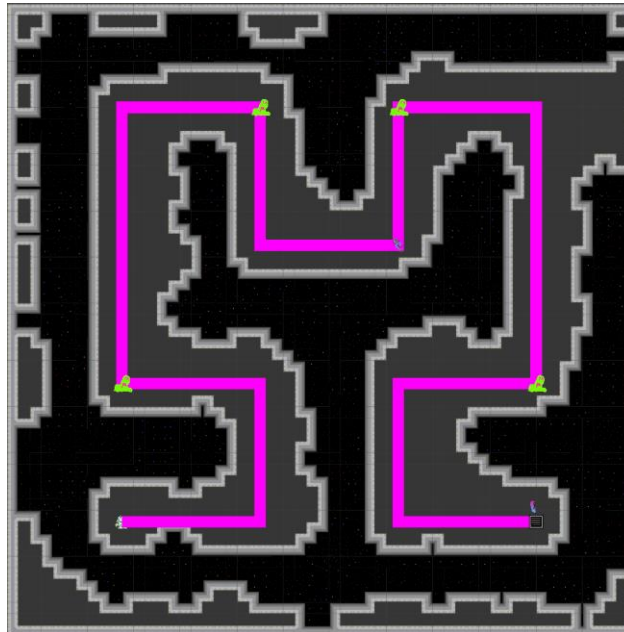


Figura 29: Curva de Moore sobre el nivel 1.

Como podemos ver, los ejemplos anteriores siguen la forma de la curva de Peano de orden 2, que es la curva fractal que se aplica en el segundo nivel. Como el tamaño del mapa es ligeramente superior al del primer nivel, la generación de la curva varía bastante más entre generaciones, dando lugar a caminos que no se:



Figura 30: Curva de Peano sobre el nivel 2.

Como podemos ver, los ejemplos anteriores siguen la forma de la curva de Sierpinski de orden 2, que es la curva fractal que se aplica en el tercer nivel:

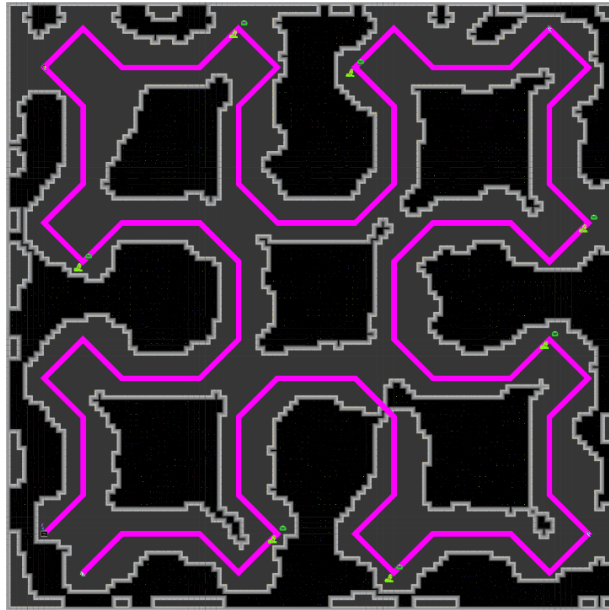


Figura 31: Curva de Sierpinski sobre el tercer nivel

Analizaremos más a fondo los elementos de un nivel, cogiendo de ejemplo el nivel 3:

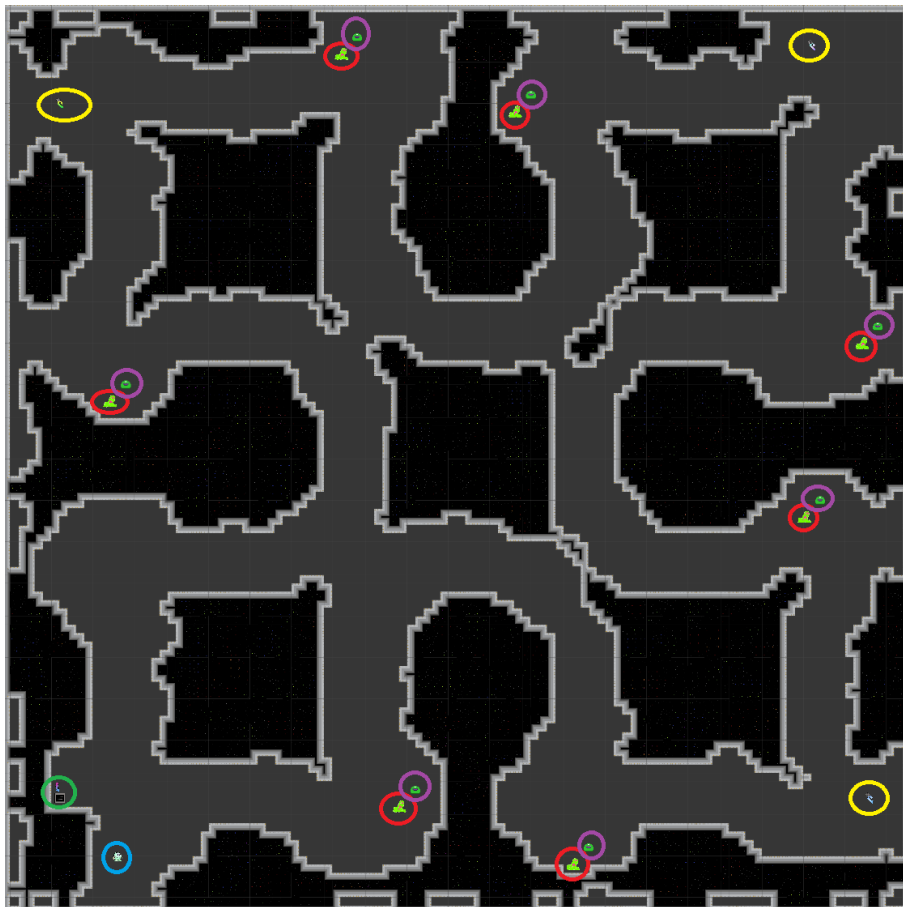


Figura 32: Elementos de un nivel.

Como cada vez el terreno generado es diferente, se instancian los diferentes elementos (protagonista, llaves, enemigos y la salida) según los puntos de la curva de cada nivel. Por tanto, todos los niveles tienen la siguiente estructura:

En primer lugar, marcado en azul, tenemos al personaje protagonista, que aparecerá siempre en la esquina inferior izquierda, en el primer punto de la curva del nivel.

A continuación, en rojo, podemos ver a al primer tipo de enemigos. Para cada nivel, se genera un diferente número de enemigos. Este enemigo aparece en los niveles uno y tres. En el caso del nivel 1, se generan 4 a lo largo de los puntos de la curva y en el tercero se generan 7.

Por otro lado, en lila, podemos ver al segundo tipo de enemigos. Estos aparecen en el nivel dos y tres. En el nivel dos se generan 10 y en el tercero 7.

En amarillo, podemos ver los coleccionables que se tendrán que recolectar antes de poder pasar al siguiente nivel. Su punto de aparición varía según el nivel, apareciendo una en el punto intermedio de la curva si es nivel 1, en el primer y tercer cuarto de la curva para el nivel 2, y tres en caso del nivel 3, en el punto intermedio y en el primer y tercer cuarto de la curva.

Por último, en verde podemos ver la salida del nivel. Esta siempre se generará al final de la curva.

Una vez alcanzada la salida con los coleccionables y sin haber perdido los 3 puntos de salud, se pasará al siguiente nivel.

Al superar los tres niveles sin perder los puntos de salud, se mostrará la pantalla de victoria y fin del juego.

Es importante mencionar que, debido a las encuestas realizadas a diversos usuarios que probaron el juego, se ha realizado un nivel de tutorial. En este, se presentan diferentes obstáculos y se muestran videos de fondo, mostrando como se superan, con tal de que el jugador comprenda las diferentes herramientas que tiene para avanzar por los niveles.

Una vez se han explicado todas, se le propone buscar una llave y llegar a la salida, teniendo que evitar a cuatro enemigos del nivel 1 y uno del nivel 2, de modo que también pueda ver como interactúan con el entorno.

Finalmente, el jugador debe llegar a la salida, como en cualquier nivel.

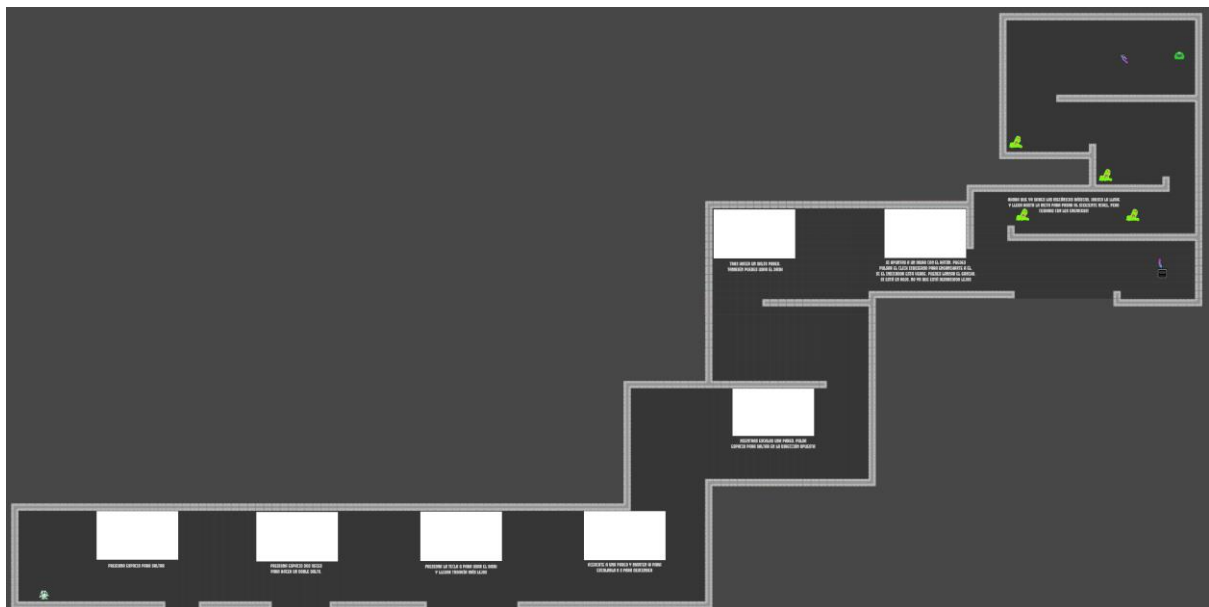


Figura 33: Tutorial del juego.

4.7. Guion / Sinopsis

Fractals Runaway cuenta una historia que sucede en un futuro lejano en el que los seres humanos, tras años de investigación, por fin logran dominar los viajes espaciales.

Una de las naves se embarca en una misión de exploración, con un equipo de cientos de humanos, con tal de investigar un planeta que aparentemente está deshabitado para poder llegar a colonizarlo.

Después de varios años de viaje, logran alcanzar dicho planeta. La tripulación consigue aterrizar y explorar su superficie, momento en el que encuentran unas viejas ruinas de civilizaciones pasadas, que escondían en ellas seres momificados.

Tras discutirlo, la tripulación decide que se deberían transportar a la nave para investigarlos al regresar a la Tierra.

Todo andaba bien hasta que, al salir de la órbita, las diferentes momias alienígenas vuelven a la vida, causando el caos en la nave y acabando con toda la tripulación, a modo de venganza por profanar sus lugares sagrados. Ahora, los extraños seres van rumbo a la Tierra para acabar con el planeta, pero por suerte, un mecanismo de alarma hace que un pequeño robot de reparaciones despierte, con la misión de alcanzar una capsula de escape y avisar así a la humanidad del peligro que se cierne sobre ellos.

Nuestro robótico amigo deberá evitar a los diferentes seres que se interpondrán en su camino. Y no solo eso, ya que, como mecanismo de seguridad de la nave, las salas previas a las escotillas de escape se generan aleatoriamente cada vez que se accede a ellas, siguiendo los patrones de diferentes curvas fractales.

Así, nuestro protagonista deberá sortear los obstáculos, encontrar las llaves para superar las salas y conseguir así escapar de la nave para advertir a los seres humanos.

5. Implementación

5.1. Requisitos de instalación

En función del sistema operativo, se requerirán diferentes características (39) :

- Windows:
 - Windows 7, Windows 10 o Windows 11.
 - Procesador con arquitectura x64 o x86.
 - GPU compatible con DX10, DX11 o DX12.
- Mac
 - High Sierra 10.13 o superior.
 - Procesador Apple Silicon con arquitectura x64
 - GPU Intel o AMD que soporten Metal
- Linux
 - Ubuntu 20.04, Ubuntu 18.04 y CentOS 7
 - Procesador con arquitectura x64
 - GPU OpenGL3.2+ que soporte Vulkan

5.2. Instrucciones de instalación

Fractal Runaway no requiere de ninguna instalación previa, se puede ejecutar directamente desde la carpeta del juego, haciendo doble clic sobre el ejecutable “FractalRunaway.exe”.

5.3. Instrucciones del juego

Para jugar a Fractal Runaway, se necesitarán pantalla, ratón y teclado. Una vez ejecutado el juego y estando en el menú principal, se podrán acceder a los diferentes menús de controles, como jugar, créditos y opciones.

Para comenzar a jugar, únicamente se deberá pulsar el botón ‘Jugar’ en la pantalla de inicio. Se recomienda que, si es la primera vez que juegas a Fractal Runaway, se haga el tutorial para entender los controles y poder disfrutar de la experiencia completa, ya que esta cuenta tanto con videos como textos que explican el funcionamiento de las diferentes habilidades disponibles.

Durante el juego, se emplean las teclas A y D para moverse horizontalmente, W y S para subir o bajar un muro, Q para usar el desplazamiento y el clic izquierdo para desplazarnos a la pared, siempre que estemos apuntando a esta con el ratón previamente. Esta información, se encuentra en la pestaña 'controles' del menú principal.

Una vez estemos jugando, se deberán encontrar las diferentes llaves de cada nivel antes de poder pasar al siguiente. Durante el juego, se podrán ver, en la esquina superior derecha, los puntos de vida restantes. En la esquina superior izquierda, se podrán ver las llaves obtenidas hasta el momento, y en la esquina inferior derecha, dos indicadores de enfriamiento del gancho y el desplazamiento, respectivamente.

A lo largo del juego, se deberán evitar a los dos tipos de enemigo, uno que simplemente se mueve de lado a lado, y otro que persigue al jugador mientras este esté dentro de un rango. Cabe destacar que los enemigos no pueden ser derrotados de ningún modo, deben esquivarse.

Al superar cada nivel, y en caso de que se cuenten con menos de tres corazones, el jugador recuperará uno.

Por último, presionando la tecla 'esc' mientras estemos en uno de los niveles, se podrá volver al menú principal y salir del juego. Además, se incluye un botón de 'regenerar nivel', que permite volver a empezar el nivel, en caso de que, debido a la generación procedural, exista algún obstáculo que no pueda superarse.

6. Demostración

6.1. Prototipos

6.1.1. Prototipos Lo-Fi

Se muestra el prototipo Lo-Fi del proyecto:

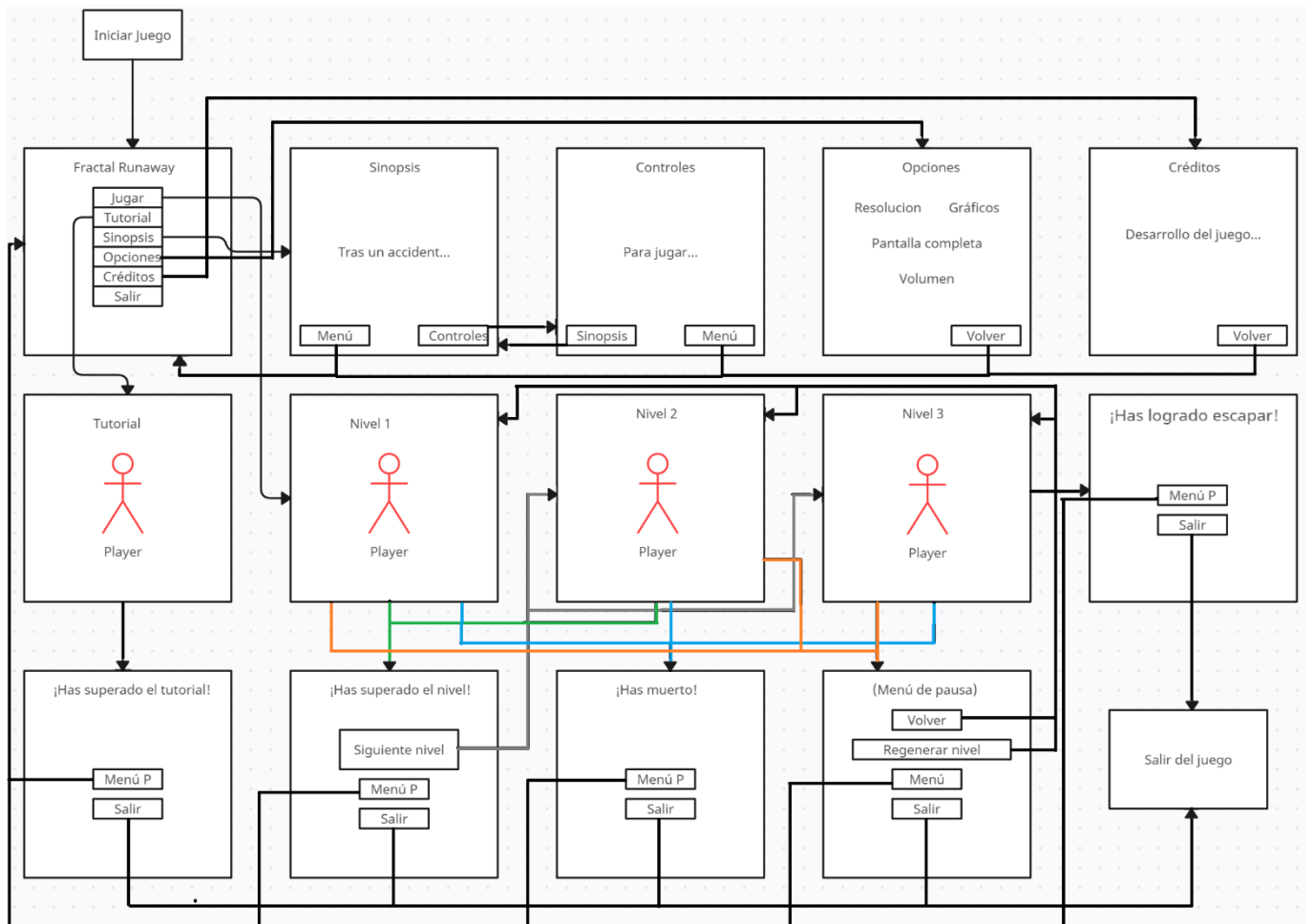


Figura 34: Prototipo Lo-Fi. (36)

6.1.2. Prototipos Hi-Fi

Para desarrollar el prototipo Hi-Fi de mi proyecto, he utilizado PowerPoint. En este, se encuentran un total de 13 diapositivas donde se muestran los diferentes menús y niveles del juego.

Se adjunta en el anexo “Entregables del proyecto”.

6.2. Pruebas de usuario

Para las pruebas del juego, he decidido contar con diferentes usuarios (amigos, familia) con tal de obtener *feedback*, poder detectar posibles errores en el desarrollo y posibles elementos a mejorar o adaptar.

En total, han realizado las pruebas 7 usuarios, que tienen las siguientes características:

- Edad: Entre 22 y 38 años.
- Todos están familiarizados con el mundo de los videojuegos, en mayor o menor medida.

6.2.1. Pruebas realizadas

Debido a la disponibilidad de los usuarios que realizaban las pruebas, las pruebas se han llevado a cabo en un plazo de dos días en total, ya que no se pudo coincidir.

Antes de la realización de las pruebas, se contactó con todos y cada uno de los usuarios, dándoles información sobre el videojuego, y se les explicó el objetivo de este, los obstáculos que podrían encontrarse a lo largo de los niveles y las habilidades que tienen para superar dichos obstáculos. Con esto, se dio rienda suelta a que cada usuario utilizase los medios proporcionados de modo que superasen los niveles a su manera, empleando los intentos que les fueran necesarios.

Finalmente, tras haber probado el juego, se envió a los usuarios un formulario de Google para que diesen su *feedback*.

Tras el primer formulario, se corrigieron ciertos errores y se añadieron los recursos gráficos definitivos, por lo que se realizó una segunda prueba, acompañada de otro formulario.

6.2.2. Análisis del formulario

Ambos formularios fueron creados mediante la herramienta Google Forms. (37) El primero, contenía 11 preguntas, y lo respondieron 4 de los 7 usuarios. A continuación, analizaré las preguntas más significativas.

- ¿Cómo valorarías tu experiencia con el juego, siendo 1 muy negativa y 5 muy positiva?
 - Dos de los usuarios respondió 5 (muy positiva) y el otro 3 (neutra).
- ¿Has conseguido superar los tres niveles?
 - Todos los usuarios respondieron “No”.
- ¿Has perdido todos los puntos de salud alguna vez?
 - Todos los usuarios respondieron “Sí”.
- ¿Cómo valorarías la dificultad del juego, siendo 1 muy sencillo y 5 muy difícil?
 - Tres de los usuarios votaron 4, y el restante, 5.
- ¿Cómo valorarías la comodidad de los controles del juego, siendo 1 muy incómodos y 5 muy cómodos?
 - Dos de los jugadores respondieron 2 (incomodos), uno respondió 3 (neutro), y el último, 4 (cómodos).
- ¿Consideras que la duración del juego es la adecuada?
 - Todos los jugadores respondieron “Sí”.
- ¿Te has encontrado con algún error a lo largo de la partida?
 - Tres de los usuarios respondieron sí, el otro, no.

Tras el cuestionario, corregí ciertos errores que los usuarios me dejaron en la pregunta “¿Hay algo que crees que pueda mejorarse?”, como diferentes bugs de los enemigos y del personaje. Además, implementé un nivel de prueba que cumple la función de tutorial, con videos explicativos y que ayuda al jugador a entender el objetivo principal del juego.

Tras estos cambios, realicé pruebas de nuevo, esta vez con 5 usuarios (dos de los usuarios ya habían respondido al formulario anterior, los otros tres eran nuevos jugadores). Al segundo formulario, le añadí dos preguntas adicionales.

Las respuestas fueron las siguientes:

- ¿Cómo valorarías tu experiencia con el juego, siendo 1 muy negativa y 5 muy positiva?
 - Dos de los usuarios respondieron 5 (muy positiva), otros dos votaron 4 (positiva) y el último, 3 (neutra).
- ¿Has conseguido superar los tres niveles?
 - 4 de los usuarios votaron “No”, y el restante votó “Sí”.
- En caso de no haber superado los 3 niveles, ¿hasta dónde has llegado?
 - De los 4 jugadores que no superaron los tres niveles, tres llegaron hasta el nivel 2, y el último hasta el nivel 3.
- ¿Has perdido todos los puntos de salud alguna vez?
 - Todos los usuarios respondieron “Sí”.
- ¿Cómo valorarías la dificultad del juego, siendo 1 muy sencillo y 5 muy difícil?
 - Dos usuarios votaron 3 (neutro), y tres votaron la opción 4 (difícil).
- ¿Cómo valorarías la comodidad de los controles del juego, siendo 1 muy incómodos y 5 muy cómodos?
 - Un usuario votó 1 (muy incómodos), otro votó 2 (incómodos), dos votaron 3 (neutro) y el último votó 4 (cómodos).
- ¿Consideras que la duración del juego es la adecuada?
 - Todos los jugadores respondieron “Sí”.
- ¿Te has encontrado con algún error a lo largo de la partida?
 - Tres de los usuarios respondió sí, el restante, no.

De los dos cuestionarios, saqué las siguientes conclusiones y realicé los siguientes cambios:

- La dificultad del juego es demasiado elevada en el nivel 2.
- Los controles son incómodos / confusos.
- La duración del juego es la adecuada.
- La curva de aprendizaje puede ser demasiado elevada, y la pantalla de explicación de controles no queda clara totalmente.
- La idea del juego ha gustado, por lo general.
- En cuanto a los gráficos añadidos para el segundo formulario, han gustado mucho a los usuarios.

Debido a esto, llevé a cabo los siguientes cambios.

- Se implementa un nivel tutorial, con videos y explicaciones de cada una de las acciones que puede llevar a cabo el personaje protagonista. Además, se explica el propósito del juego, teniendo que recoger una moneda y llegar hasta el final, evitando a cuatro enemigos.

7. Conclusiones y líneas de futuro

7.1. Conclusiones

7.1.1. Lecciones aprendidas

A lo largo del desarrollo, se han adquirido conocimientos en un gran número de campos y se han ampliado otros con los que ya contaba:

- Se ha aprendido a manejar con fluidez la herramienta de Unity, siendo capaz de realizar todo tipo de tareas.
- Se ha profundizado en la gestión de un proyecto real, así como las metodologías de trabajo aplicadas.
- Se ha aprendido el funcionamiento práctico de un autómata celular, así como los ajustes para obtener diversos resultados en función de sus parámetros.
- Se ha aprendido sobre las curvas fractales y los algoritmos que los generan, de las cuales apenas sabía nada antes de comenzar el proyecto.
- Se ha ganado soltura en el uso de la herramienta de gestión de código 'GitLab'.

7.1.2. Objetivos

Los objetivos propuestos en los puntos 1.3, se han conseguido todos:

- Se han creado tres niveles que permiten el uso de las tres mecánicas.
- Se han conseguido generar los escenarios de los tres niveles empleando un autómata celular, y aplicando a cada nivel una curva fractal diferente, de modo que, dentro de la aleatoriedad, los niveles tengan un sentido.
- Se ha creado una trama para el juego, que describe al protagonista, el mundo en el que sucede el juego y la motivación.
- Se han creado, además de las mecánicas de movilidad básicas, las dos mecánicas adicionales planteadas en los objetivos, el gancho y el deslizamiento, con tal de moverse con mayor fluidez por el escenario.
- A nivel personal, se han adquirido una gran cantidad de conocimiento sobre el mundo del videojuego, el desarrollo de proyectos y de las diferentes herramientas empleadas.
- Por último, se han conseguido los recursos gráficos y de audio, proporcionados por dos amigos que han querido colaborar en esos aspectos.

7.1.2 Planificación

En cuanto a la planificación, se han seguido los tiempos estipulados en el diagrama de Gantt del punto 1.5 de la memoria, cumpliendo los diferentes hitos, entregando en cada PEC el contenido deseado. Ha habido tareas, como la creación del nivel 2 y 3, que han llevado menos tiempo del esperado, pues se ha reutilizado código, teniendo únicamente que implementar la curva fractal correspondiente. No obstante, otras tareas, como el diseño de los enemigos o las diferentes habilidades del protagonista, han requerido más dedicación.

Se han respetado las diferentes fases de la metodología agile, cumpliendo en cada iteración con el objetivo de esta (diseño, desarrollo, pruebas...). La considero una metodología óptima, que permite adaptarse a las circunstancias y centrarse en aquello en lo que se está trabajando en cada momento, pudiendo volver a hacer iteraciones sobre el trabajo ya hecho en caso de necesitar llevar a cabo cualquier modificación.

7.2. Líneas de futuro

Como ya menciono en diferentes puntos de esta memoria, este proyecto me ha servido como introducción al mundo de los videojuegos. Mi objetivo una vez haya acabado la carrera, es comenzar un máster en desarrollo y programación de videojuegos.

A pesar de que el juego no recibirá actualizaciones ni se comercializará, los conocimientos adquiridos a lo largo del desarrollo han sentado una base sólida en el mundo de la creación de videojuegos.

No obstante, hay ciertos elementos que sí me gustaría implementar con tal de poder aprender más, y que no se han llevado a cabo por falta de tiempo, como un registro de los datos de juego, o una comparativa entre jugadores, para ver quien lo completa más rápidamente.

8. Bibliografía

1. *MDA: A Formal Approach to Game Design and Game Research*. **Hunicke, Robin, LeBlanc, Marc y Zubek, Robert**. 2004.
2. **Technologies, Unity**. Unity. [En línea] 2005. <https://unity.com/es>.
3. *The general and logical theory of automata*. **von Neumann, John**. 1951, Cerebral Mechanisms in Behavior – The Hixon Symposium, págs. 1-31.
4. **Tricot, Claude**. *Curves and Fractal Dimension*. New York : Springer-Verlag, 1995.
5. **GanttPRO**. GanttPro. [En línea] 2014. <https://app.ganttpro.com/#/>.
6. **Glassdoor**. Guionista. *Sueldo para el puesto de guionista en España*. [En línea] 1 de enero de 2023. https://www.glassdoor.es/Sueldos/guionista-sueldo-SRCH_KO0,9.htm.
7. —. Programador. *Sueldos para el puesto de programador FullStack en España*. [En línea] 2023 de abril de 2023. https://www.glassdoor.es/Sueldos/programador-fullstack-sueldo-SRCH_KO0,21.htm.
8. —. Artista 2D. *Sueldos para el puesto de Artista 2D en España*. [En línea] 14 de abril de 2023. https://www.glassdoor.es/Sueldos/artista-2d-sueldo-SRCH_KO0,10.htm.
9. —. Animador 2D. *Sueldos para el puesto de animador 2D en España*. [En línea] 21 de febrero de 2023. https://www.glassdoor.es/Sueldos/animador-2d-sueldo-SRCH_KO0,11.htm.
10. —. Técnico de sonido. *Sueldos para el puesto de técnico de sonido en España*. [En línea] 5 de abril de 2023. https://www.glassdoor.es/Sueldos/t%C3%A9cnico-de-sonido-sueldo-SRCH_KO0,17.htm.
11. —. Analista de marketing. *Sueldos para el puesto de analista de marketing en España*. [En línea] 19 de abril de 2023. https://www.glassdoor.es/Sueldos/analista-de-marketing-sueldo-SRCH_KO0,21.htm.
12. **Glassdoor**. Glassdoor. [En línea] 2007. <https://www.glassdoor.es/index.htm>.
13. **AEVI, Asociación Española de Videojuegos**. La industria del videojuego en España en 2021. [En línea] 2021. http://www.aevi.org.es/web/wp-content/uploads/2022/04/AEVI_Anuario_2021_Final.pdf.
14. **Weisstein, Eric W**. "von Neumann Neighborhood". *MathWorld--A Wolfram Web Resource*. [En línea] <https://mathworld.wolfram.com/vonNeumannNeighborhood.html>.
15. —. "Moore Neighborhood.". *MathWorld--A Wolfram Web Resource*. [En línea] <https://mathworld.wolfram.com/MooreNeighborhood.html>.
16. **Weisstein, Eric W**. Wolfram MathWorld. [En línea] 1995. <https://mathworld.wolfram.com/>.
17. *Über die stetige Abbildung einer Linie auf ein Flächenstück*. **Hilbert, David**. 1891, Mathematische Annalen, págs. 459-460.
18. *Sur une courbe continue sans tangente, obtenue par une construction géométrique élémentaire*. **koch, Helge**. 1904, Arkiv för matematik, págs. 681-704.
19. *Sur une courbe, qui remplit toute une aire plane*. **Peano, Giuseppe**. 1890, Mathematische Annalen, págs. 157-160.

20. **Rodríguez Santos, Alberto.** Epsilones. [En línea] 2022.
<https://www.epsilones.com/paginas/historias/historias-008-antes-mandel.html>.
21. **Aikman, Z.** Youtube - Unite 2014 - Generating Procedural Dungeons in Galak Z. [En línea] 26 de Agosto de 2014. https://www.youtube.com/watch?v=ySTpjT6JYFU&t=2029s&ab_channel=Unity.
22. *Improving procedural 2D map Generation based on multi-layered cellular.* **Macedo, Yuri P. A. y Chaimowicz, Luiz.** 2017, 2017 16th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames), págs. 116-125.
23. **Pav.** PavCreations. *Procedural generation of 2D maps in Unity.* [En línea] 07 de 02 de 2022.
<https://pavcreations.com/procedural-generation-of-2d-maps-in-unity/>.
24. *Mathematical models for cellular interactions in development II. Simple and branching filaments with two-sided inputs.* **Lindenmayer, Aristid.** 1968, Journal of Theoretical Biology, págs. 300-315.
25. *On certain crinkly curves.* **Moore, E. H.** 1900, Trans. Amer. Math. Soc, págs. 72-90.
26. *Utilization of Fractal Geometry for Phase Shifter Implementation.* **Chakraborty, A. y Kar, A. K.** 2016, Proceedings of the International Conference on Signal, Networks, Computing, and Systems, págs. 317-323.
27. *A Peano fractal-based dual-mode microstrip bandpass filters for wireless communication systems.* **Ali, J. K., y otros.** 2012, Proceedings of Progress in Electromagnetics Research Symposium.
28. *Sierpiński Curve.* **Weisstein, Eric W.** 2019, MathWorld.
29. **Canva.** Canva. [En línea] 2013. https://www.canva.com/es_es/.
30. **Udemy.** Udemy. *Complete C# Unity Game Developer 2D.* [En línea] 2019.
<https://www.udemy.com/course/unitycourse/>.
31. **Microsoft.** Microsoft365. [En línea] 1975. <https://www.microsoft.com/es-es/microsoft-365>.
32. **B.V., GitLab.** GitLab. [En línea] 2011. <https://about.gitlab.com/>.
33. **Services, Logitech.** Streamlabs. [En línea] 2011. <https://streamlabs.com/es-es>.
34. **Labs, Notion.** Notion. [En línea] 2016. <https://www.notion.so/>.
35. **Pixilart.** Pixilart. [En línea] 2015. <https://www.pixilart.com/draw#>.
36. **Cinergix.** Cretely. [En línea] 2008. <https://creately.com/plans/?ref=home>.
37. **Google.** Google Forms. [En línea] 2018. <https://docs.google.com/forms/u/0/>.
38. **Ventura, Alex.** LinkTree. [En línea] 2023. <https://linktr.ee/Mekanomicon>.
39. **Unity.** System requirements for Unity 2021 LTS. *Unity Player system requirements.* [En línea] 2021. <https://docs.unity3d.com/Manual/system-requirements.html>.
40. **Ware, Bryan.** Pixilart. [En línea] 2024. <https://www.pixilart.com/>.
41. *Procedural game level generation by joining geometry with hand-placed connectors.* **Silva, R. C., y otros.** 2020.
42. **Boris.** BorisTheBrave. *Dungeon Generation in Binding of Isaac.* [En línea] 12 de 09 de 2020.
<https://www.boristhebrave.com/2020/09/12/dungeon-generation-in-binding-of-isaac/>.
43. —. BorisTheBrave. *Dungeon Generation in Enter The Gungeon.* [En línea] 28 de 07 de 2019.
<https://www.boristhebrave.com/2019/07/28/dungeon-generation-in-enter-the-gungeon/>.

44. *ANÁLISIS CARTOGRÁFICO DE LA DIMENSIÓN FRACTAL PARA LA VARIACIÓN ESPACIO-TEMPORAL DE RÍOS.* **Fernández, H. L. G., Cortés, L. A. y González, M. Á.** 2019, Revista Geoespacial,, págs. 78-97.

Anexos

Anexo A: Entregables del proyecto



Fractal Runaway
HiFi.pptx

Anexo B: Resultados detallados de una encuesta



Pruebas de Fractal
Runaway.pdf



Segunda encuesta
Fractal Runaway.pdf
