

Creación procedural básica. Creación de un videojuego de forma procedural y comparativa.

Autor: Enrique Cano Rojas.

Tutor: Helio Tejedor Navarro.

Profesor: Joan Arnedo Moreno

Master en Programación y Diseño de Videojuegos

Diseño de experiencias de juego

06/05/2022

Créditos/Copyright



Esta obra está sujeta a una licencia de Reconocimiento- No Comercial - Sin Obra Derivada

[3.0 España de Creative Commons.](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Creación procedural básica. Creación de un videojuego de forma procedural y comparativa.</i>
Nombre del autor:	<i>Enrique Cano Rojas</i>
Nombre del colaborador/a docente:	<i>Helio Tejedor Navarro Jordi Duch Gavaldá</i>
Nombre del PRA:	<i>Joan Arnedo Moreno</i>
Fecha de entrega (mm/aaaa):	<i>06/2022</i>
Titulación o programa:	<i>Master en Programación y Diseño de Videojuegos</i>
Área del Trabajo Final:	<i>Diseño de experiencias de juego</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>3D, Procedural, Isométrico</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo</i>	
<p>En este proyecto se procederá a crear un videojuego de carácter de estudio sobre la generación procedural y el efecto que tiene sobre la generación de mundos. El cual se basará en la generación procedural como punto clave. Se establecerá con la finalidad de crear una experiencia original una cámara isométrica y un escenario acorde a la premisa del juego. Al ser un proyecto individual no se ha establecido como un juego extenso, ni tampoco de gran calidad grafica. Por consiguiente, la metodología que se ha seguido ha sido un proceso en cascada donde se iniciara con la creación de personaje, así como de la generación procedural seguido de la creación de enemigos básicos y una dungeon. Como último paso se creará una HUD básica y un menú. Con la finalidad de facilitar el proyecto se ha hecho uso de los ScriptableObject como herramienta fundamental. Como resultado se ha creado una dungeon de tamaño medio modificable usando una serie de parámetros básicos, modificables. En conclusión, del proyecto se establece que las posibilidades al usar código procedural mejoran considerablemente la variabilidad en las que las experiencias de cada usuario, a su vez el uso de los ScriptableObjects como pilar central permite una variabilidad establecida. El proyecto a su vez da paso a ser extensible a largo plazo.</p>	
Abstract (in English, 250 words or less):	

In this project, we will proceed to create a video game of a study nature about procedural generation and the effect it has on the generation of worlds. Which will be based on procedural generation as a key point. An isometric camera and stage will be set according to the game's premise to create an original experience. Being an individual project, it has not established itself as an extensive game, nor one of great graphic quality. Therefore, the methodology that has been followed has been a cascading process that began with the creation of characters, as well as the procedural generation of the world followed by the creation of basic enemies. As a last step, a basic HUD and menu will be created. To facilitate the project, the ScriptableObject has been used as a fundamental tool. As a result, a medium-sized modifiable dungeon has been created using a number of basic modifiable parameters. In conclusion, the project establishes that the possibilities of using procedural code considerably improve the variability in which each user experiences, in turn the use of ScriptableObject as a central pillar allows to establish variability. Finally, the project in turn gives way to being extensible in the long term.

Abstract

En este proyecto se procederá a crear un videojuego de carácter de estudio sobre la generación procedural y el efecto que tiene sobre la generación de mundos. El cual se basará en la generación procedural como punto clave. Se establecerá con la finalidad de crear una experiencia original una cámara isométrica y un escenario acorde a la premisa del juego. Al ser un proyecto individual no se ha establecido como un juego extenso, ni tampoco de gran calidad grafica. Por consiguiente, la metodología que se ha seguido ha sido un proceso en cascada donde se iniciara con la creación de personaje, así como de la generación procedural seguido de la creación de enemigos básicos y una dungeon. Como último paso se creará una HUD básica y un menú. Con la finalidad de facilitar el proyecto se ha hecho uso de los ScriptableObject como herramienta fundamental. Como resultado se ha creado una dungeon de tamaño medio modificable usando una serie de parámetros básicos, modificables. En conclusión, del proyecto se establece que las posibilidades al usar código procedural mejoran considerablemente la variabilidad en las que las experiencias de cada usuario, a su vez el uso de los ScriptableObjects como pilar central permite una variabilidad establecida. El proyecto a su vez da paso a ser extensible a largo plazo.

In this project, we will proceed to create a video game of a study nature about procedural generation and the effect it has on the generation of worlds. Which will be based on procedural generation as a key point. An isometric camera and stage will be set according to the game's premise to create an original experience. Being an individual project, it has not established itself as an extensive game, nor one of great graphic quality. Therefore, the methodology that has been followed has been a cascading process that began with the creation of characters, as well as the procedural generation of the world followed by the creation of basic enemies. As a last step, a basic HUD and menu will be created. To facilitate the project, the ScriptableObject has been used as a fundamental tool. As a result, a medium-sized modifiable dungeon has been created using a number of basic modifiable parameters. In conclusion, the project establishes that the possibilities of using procedural code considerably improve the variability in which each user experiences, in turn the use of ScriptableObject as a central pillar allows to establish variability. Finally, the project in turn gives way to being extensible in the long term.

Palabras clave

ScriptableObject, Procedural, Isométrico

Índice

1.	Introducción.....	9
1.1.	Introducción/Prefacio.....	9
1.1.1.	Historia	9
1.2.	Descripción/Definición.....	10
1.3.	Objetivos generales.....	11
1.3.1.	Objetivos principales.....	11
1.3.2.	Objetivos secundarios	11
1.4.	Metodología y proceso de trabajo.....	12
1.5.	Planificación.....	13
1.5.1	Tabla	14
1.6.	Presupuesto.....	16
1.6.1.	Equipo	16
1.6.2.	Monetización	16
2.	Análisis de mercado	17
2.1.	Público objetivo (i.e. <i>target audience</i>) y perfiles de usuario.....	17
2.2.	Competencia/Antecedentes (estado del arte).....	18
2.3.	Análisis DAFO	19
3.	Propuesta	21
3.1.	Definición de objetivos/especificaciones del producto.....	21
3.2	Estrategia de marketing	22
4.	Diseño	23
4.1.	Arquitectura general de la aplicación	24
4.2.	Arquitectura de la información y diagramas de navegación	24
4.3.	Diseño gráfico e interfaces.....	26
4.3.1.	Estilos.....	26
4.3.2.	Usabilidad /UX	27
4.4.	Lenguajes de programación y APIs utilizados	28
5.	Implementación	29

5.1. Instrucciones de instalación	29
6. Demostración.....	30
6.1. Instrucciones de uso	30
6.2. Prototipos	30
6.2.1. Prototipos Lo-Fi.....	30
6.2.2. Prototipos Hi-Fi	31
6.3. Ejemplos de uso del producto (o guía de usuario).....	32
7. Conclusiones y líneas de futuro.....	33
7.1. Conclusiones	33
7.2. Líneas de futuro	33
Bibliografía.....	35

Figuras y tablas

Índice de figuras

Figura 1: Diagrama de Gantt.....	13
Figura 2: Diagrama creado usando www.lucidchart.com ScriptableObjects.....	27
Figura 3: Diagrama creado usando www.lucidchart.com “basic system”.....	27
Figura 4: Paleta de colores básica de tonos blanco a negro.....	28
Figura 5: Paleta de colores obtenida de la página color Palletes de Adobe.....	28
Figura 6: Hipotético menú para el juego con título y botones. Creada usando Adobe Photoshop e Illustrator.....	29
Figura 7: Hipotético User Interface. Creada usando Adobe Photoshop e Illustrator e imágenes in Engine.....	29
Figura 8: Hipotético User Interface. Creada e implementada temporalmente en Unity para testeo.....	30
Figura 9: Boceto Conceptual Personaje Principal.....	32
Figura 10: Boceto Conceptual Enemigo Básico.....	32
Figura 11: Boceto Conceptual Visión del Mundo.....	33
Figura 12: MockUp Creado a partir de imágenes de varias páginas de internet.....	33
Figura 1: MockUp 2 Estilos de juego similares, dungueon crawler y otros.....	34

1.Introducción

1.1. Introducción/Prefacio

Se pretende crear como producto un juego Rogué Like para la plataforma de PC. La reciente necesidad de crear un juego que dé, decisiones al jugador, a costa de algo a cambio. En otros juegos comparables, como puede ser “The binding of Isaac” la dificultad viene dada principalmente por las decisiones y el riesgo que quiera tomar cada jugador a la hora de empezar la partida, así como la ruta hacia el final. Si se elige un camino corto las posibilidades de éxito se reducen mientras que si se elige un camino largo la posibilidad de alcanzar el final se incrementarían dependiendo del tipo de objetos que este adquiriera durante el camino. La idea original de este proyecto era crear un Rogué Like de dificultad ascendente dependiendo de la altura, es decir el jugador podría llegar más rápido al final si desciende de un piso de forma abrupta, pero perdería parte de lo que podría encontrar en ese nivel, además de que el incremento de dificultad se incrementaría bastante más. Tomando como punto de referencia otros juegos como “Neon Abyss”, cambiando la verticalidad a 3D.

1.1.1. Historia

“El Cielo se cae y el protagonista con él” basándose en esto, los dioses del Valhala han elegido a un juguete o peón para jugar con el mientras da inicio el Ragnarok, fragmentos de realidad están cayendo hacia la tierra conectados por pasillos usando las raíces del árbol Yggdrasil el jugador podrá pasar de habitación en habitación eliminando a los enemigos que se encontrará a su paso. Entre tanto las bestias del Ragnarok se opondrán al jugador como bosses por las fuerzas del inframundo que se han escapado, así como vikingos, harán de enemigos.

La historia pretende usar a un personaje genérico del mundo moderno/actual vestido con ropajes que le dan poder otorgados por los dioses. Por ende, tendera a ser simplista su diseño, para que sea lo más familiar y distinguible posible.

1.2. Descripción/Definición

Se pretende crear un Rogué Like estilo clásico, en unity con una temática diferente a lo usual. La necesidad de crear uno de este estilo surge de que últimamente los Rogué Like están siendo explorados en todas sus vertientes, desde su auge con la salida de Hades (2018), muchos se han combinado con otros géneros para dar una vuelta de hoja a las mecánicas, véase como ejemplos: "Neon Abyss", "Risk of Rain 1", "Risk of Rain 2", "Loop hero", "Slay the Spire", "Monster Train", etc... Por esto mismo se pretende explorar hasta qué punto la generación procedural puede afectar a crear distintas sensaciones a los usuarios, generando 7 habitaciones de forma manual como algo básico. Como punto de partida se tomarán salas cuadradas de determinado tamaño en unidades estimadas de 20 x 20 en Unity y a partir de ahí se diseñarán los movimientos básicos del personaje, enemigos y se desarrollara el sistema de combate. Por otra parte, secundariamente, se estudiarán los sistemas de generación procedural para ver su efecto en este tipo de juegos y el efecto que tienen sobre el jugador. Crear además se creará un sistema de enemigos, basado en el número de habitaciones y el nivel de jugador.

El Trabajo se enfocará en un "Rogué Like" en 2.5D, en vista isométrica pero paralela, estilo aproximadamente similar a "One Step from Eden", basada en 45° de movimiento. Posibles usos de sprites en escenarios en 3 Dimensiones, posibles enemigos en 3D, con una cámara en perspectiva que siga al jugador. Se propone entonces, crear 7 niveles/habitaciones distintas, incluyendo una habitación de Boss, una habitación de inicio y de forma secundaria una posible habitación de comercio. Para el sistema de combate se pretende mezclar el "New Input System", con el antiguo sistema de Unity, para crear un sistema de combate manualmente y así ampliar la experiencia de usuario usando otro tipo de controles. Con vida, estamina y factores numéricos que determinen velocidad del jugador, resistencia, daño y otros.

1.3. Objetivos generales

Listado y descripción de los objetivos del TF, ordenados por relevancia.

1.3.1. Objetivos principales

Objetivos de la aplicación/producto/servicio:

- Producir una experiencia satisfactoria en el sistema de combate.
- Crear una experiencia única en cada inicio de partida.
- Crear una comparativa entre producto generado de forma procedural y producto generado manualmente.
- Crear un sistema de control único para la aplicación.

Objetivos para el cliente/usuario:

- Disfrutar de algo distinto a lo que usualmente están teniendo los Rogué Like.
- Divertirse, explorando un entorno 3D con un personaje en 2D.

Objetivos personales del autor del TF:

- Desarrollar un juego aleatorio a mayor escala.
- Posible venta del producto y desarrollo del mismo a posteriori.
- Experiencia publicando un juego.
- Crear una experiencia de usuario estable y satisfactoria.

1.3.2. Objetivos secundarios

Objetivos adicionales que enriquecen el TF.

- Investigar los sistemas de generación de procedural para aplicaciones.
- Investigar y explorar los sistemas de control de juego. "New Input System"
- Investigar sistemas creados en código para a posteriori crear un sistema de objetos basado en inventario.

1.4. Metodología y proceso de trabajo

En principio al ser un proyecto realizado por un equipo de una persona se pensó en utilizar una **metodología en cascada**, donde se llevarían procesos en paralelo, pasando de objetivo en objetivo hasta ir cumplimentando los objetivos esenciales, enfocado en la Experiencia de Usuario.

Cuyos procesos clave serán:

Análisis de las necesidades del producto -> Creación de contenido software -> Testeo del producto -> Corrección de errores.

Así a cada escalón en cascada se hará un breve testeo de lo implementado y se hará una pequeña corrección de errores para a su vez en horas específicas de depuración, eliminar o corregir código innecesario del proyecto.

A su vez se dedicarán sesiones únicas para escribir y detallar la memoria del proyecto para que quede lo más completo posible, terminando el plazo de cada uno de las etapas del mismo.

Otras metodologías que hubieran sido posibles y aptas para este proyecto podrían haber sido las metodologías Agiles como SCRUM o programación extrema (XP) y Kanban. Si el grupo de trabajo hubiese sido más amplio, pues son metodologías, creadas y diseñadas para grupos de personas y varios puestos de trabajo sobre un mismo proyecto.

1.5. Planificación

Flechas clave:

03/04/2022 Primera Entrega (Demo técnica).

09/05/2022 Segunda Entrega (Versión Jugable).

05/06/2022 Tercera Entrega (Versión Completa).

Diagrama de Gantt:

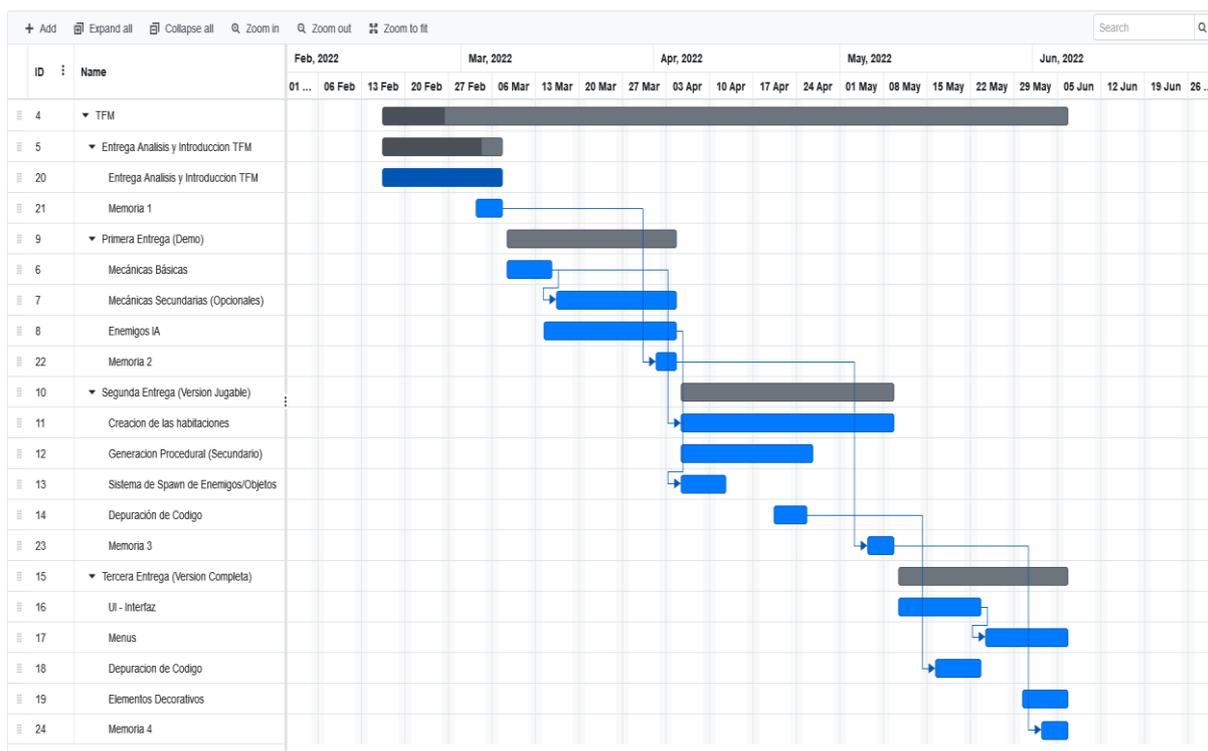


Figura 1: Diagrama de Gantt

1.5.1Tabla

Etapa	Implementación	Tiempo Estimado	Problemas Esperados	Soluciones	Tiempo/Fechas
1	Mecánicas Básicas	7 días	Problemas con la cámara y configuración para llegar al objetivo del proyecto. Se espera usar el complemento Cinemachine.	Búsqueda de Código complementario para dar profundidad a los movimientos.	Entrega 1 03/04/2022
2	Mecánicas Secundarias (Opcionales)	7 días (+Tiempo sobrante de la primera etapa)	Problemas con la mezcla de controles entre el Input System y el New Input System		Entrega 1 03/04/2022
3	Generación Procedural (secundario) // Escenario, niveles	15 días	Posibles bugs de código al ser implementado.	Si en el estimado tiempo no se puede realizar la generación procedural, se procederá a omitir este paso. Creando 7 salas predefinidas para la hora de la entrega.	Entrega 2 09/05/2022
4	Depuración de Código	5 días	Periodo para la corrección de errores pequeños y simplificación de	Se procederá a la eliminación o corrección de código que no sea deseado.	Entrega 2 09/05/2022

			código a posteriori		
5	Enemigos IA	7 días	Intentando usar el componente NavMeshAgent se planea crear 2 tipos de enemigos y un Boss con sus propios movimientos. Puede haber interferencias entre los controles o problemas al spawnear los enemigos		Entrega 1 03/04/2022
6	UI/Interfaz	7 días	Diseño de UI en base a ScriptableObject. Posible causa de errores		Entrega 3 05/06/2022
7	Menús	10 días			Entrega 3 05/06/2022
8	Sistema de Spawn de objetos/Enemigos	5 días	Posible interferencia con el NavMeshAgent que cause problemas al instanciar los enemigos		Entrega 2 09/05/2022
9	Depuración de código.	5 días			Entrega 3 05/06/2022

10	(Opcionales) Elementos decorativos.	(Resto del tiempo hasta la entrega)	Posible problema de colisiones.		Entrega 3 05/06/2022
----	---	--	---------------------------------------	--	-------------------------

1.6. Presupuesto

Combinando tiempo y material se puede presuponer que el precio por un producto así podría rondar alrededor de los 4000-6000 euros. Ya sea por el equipo informático necesario, las licencias de los productos de desarrollo: Unity, Adobe, Microsoft office, Daz Studio, Maya y otros. Los componentes de Hardware necesarios ordenador con potencia grafica para soportar el desarrollo unitario, el material extra utilitario como es el ratón, tableta gráfica y escáner. Todo forman parte del desarrollo.

1.6.1. Equipo

Equipo no Informático disponible:

- Silla de oficina
- Escritorio de trabajo

Equipo Informático disponible:

- Ordenador i7 con 16 gb de RAM 3050ti
- Escáner
- Tableta grafica
- Ratón

Programas de diseño 2D:

- Adobe Photoshop
- Adobe Illustrator

Programas de Postproducción:

- Adobe Premiere
- Adobe After Effects.

Programas de uso para la creación de videojuegos:

- Unity
- Daz Studio
- Maya

1.6.2. Monetización

Basándose en que los rogué Little están cada vez más presentes en el mundo de los juegos, se pretende crear uno con una larga vida de duración alrededor de 2-4 años. Con una monetización base

de un solo pago al inicio de 4.5 euros, y precio incremental a medida que tenga una mayor aceptación añadiendo más contenido hasta un máximo de 12-15 euros a futuro.

2. Análisis de mercado

Actualmente con la reciente salida de juegos triple A como puede ser “Elder Ring” o juegos como “Horizon Zero Dawn” juegos de mundo abierto y en tercera persona y si previsión de que salga algún juego de este estilo que opaque la salida de este proyecto. Se puede pensar que tendrá un recibimiento decente a la hora de salida de este proyecto.

2.1. Público objetivo (i.e. *target audience*) y perfiles de usuario

Siendo un producto de carácter dentro del marco del entretenimiento audiovisual y que será accesible por una cantidad considerable de personas de distintos países desde una plataforma como podría ser Steam. El lenguaje del juego durante el prototipado será en español sin embargo se planea la traducción al inglés para la versión final. Teniendo una temática centrada en mitología mezclada con algo de vida “Real” podrían enmarcarse el público potencial entre un rango de edades de 16-50 años, centrándose en un rango de edades de 16-30 años debido a que son personas con más tiempo potencial a la hora de dar tiempo a un juego. Sin embargo, se ha visto que esta tendencia esta puesta a ampliarse a futuro. Siendo el rango de edades:

- 16-25: Personas posiblemente dispuestas a partidas de 1-3 horas. **Perfil de usuario fijo:** El cual quiere exprimir el recurso al máximo.
- 25-40: Personas que posiblemente estén dispuestas a partidas de 1 hora o menos. **Perfil de usuario casual:** persona ocupada la cual no puede ofrecer todo su tiempo al juego sin embargo le resulta entretenido y puede dedicar una partida en su tiempo libre.
- 40-50: Personas que posiblemente estén dispuestas a partidas de menos de 50 minutos. Perfil de usuario hiper casual: Personas dadas a jugar algo, pero por causas personales o ajenas no puede dedicarle demasiado tiempo al juego, lo que le da un margen de tiempo bastante acotado.

2.2. Competencia/Antecedentes (estado del arte)

Como competencias a este título podemos encontrar juegos Rogué Like,

Juegos 2.5D:

- "The Binding of Isaac", "Enter the Gungeon", "Exit the Gungeon", "Neon Abyss", "One Step From Eden", "Hades"

Juegos 3D:

- "The Risk of Rain", "Curse of the Dead Gods", "The Risk of Rain 2", "Zigurat", "Zigurat 2", "Rogue Towe", "Nightmare Reaper" y otros.

Títulos con gran éxito y que han sido bastante alabados en algunos casos, a comparación de todos los títulos no se ha encontrado un título Rogué Like en 3D que se aproxime al parecido del proyecto, el único que se ha encontrado que puede tener alguna similitud con respecto a la perspectiva y comportamiento del personaje en perspectiva es "Curse of the Dead Gods" y "Hades" cuya temática y estilo se diferencia bastante del resultado que se quiere alcanzar con este título. A nivel de escenario y dungeon se puede comparar a "The Binding of Isaac" pero siendo un título casi por completo en 2D, también se diferencia bastante del resultado a alcanzar. En relación a estética no se ha encontrado ningún título que tenga parecido, en el ámbito de los Rogué Like.

En relación a lo que pretende ser especial de este proyecto es que actualmente no hay Rogué Like en 3D completamente que aprovechen las ventajas y el punto de vista que ofrece este aspecto.

Seguidamente se hará el análisis de los Pros y los contras del título.

Pros:

- Rogué Like de estilo clásico en 3D-2.5D (Aun se sopesan los Sprites en cuyo caso se intentaría a hacer algo parecido a títulos como "Paper Mario: The origami King").
- Movimiento libre 3D en 4 axis libertad de movimiento en las 8 direcciones del plano.
- Verticalidad con salto. Posible subida o bajada de niveles (a nivel del suelo).
- Título sencillo/Corto. Duración estimada 20 minutos por nivel.
- Estética "Nórdica" /Occidental: Estilo o Visualización occidental del estilo nórdico. No se requiere similitud alta. Se quiere dar un aire distinto o diferente.

Contras:

- Diseño simplista.
- Movimiento limitado, para algunas personas. A comparación de títulos como "Hades" y "Curse of the Dead Gods".
- Verticalidad limitada. A comparación de títulos como "Neon Abyss".

-Estilo de estética explotada por otros títulos ajenos. Ejemplos de títulos: “Assasins Creed: Valhalla”, “God of War (2018)”, “God of War: Ragnarok (Aun sin salida)”, “Valheim”, “Viking City Builder”, “Brawlhalla” y otros..

Ejemplos de referencias bibliográficas (Manovich, 2011) en esta frase (Anderson, 2012) (ver el apartado Bibliografía en este documento).

2.3. Análisis DAFO

Debilidades.

-Nivel Técnico. Al ser un proyecto de una persona en un corto plazo a nivel técnico de salida el juego será algo decadente. Sin embargo, se pretende reforzar a futuro. El sistema procedural que se usará para el título será simple en comparación a otros, ya que se usará a nivel comparativo contra un nivel rehecha de antemano. Lo que podría ocasionar que a nivel potencial se desperdiciase gran parte del esfuerzo en acomplejar el sistema procedural, lo que ocasionaría algo de desagrado al público potencial.

-Nivel Estético. En comparación a otros títulos carecerá de una gran estética, pero estable. Siendo un juego desarrollado por solo 1 persona se esperan limitaciones a nivel estético y otros niveles.

-Nivel Historia. Tendrá una historia simple inicialmente no enfocada en ella. Lo que causara que se deje en segundo plano inicialmente.

Fortalezas.

-Precio. Sera de bajo coste en comparativa a otros títulos inicialmente.

-Rapidez. Juego rápido para usuarios con poco tiempo. Diseñado para que a medida que se juegue se fortalezca el jugador en cada partida.

-Nivel de Diseño. De escenario y escenografía.

-Nivel de Estética. La combinación de 3D-2.5D junto al estilo de Rogué Like.

-Nivel de Jugabilidad. Único en el estilo de juego siendo en perspectiva 8 direcciones libres con verticalidad libre.

Amenazas

-Posible salida de algún título de gran calidad a la hora de la salida. Esto influiría negativamente a la hora de publicar el título, ya que se ha demostrado que la salida de grandes títulos triple A “opacan” la salida de otros títulos.

Oportunidades

-Si se perfila correctamente podría tener una gran aceptación en el momento de la salida lo que llevaría a un desarrollo del título más allá de lo que podría ofrecer este proyecto.

3.Propuesta

Se propone crear un juego estilo Rogué Like para PC de carácter comparativo, para su futura venta en la tienda de "Steam" u "Epic Store". De carácter 2.5D-3D (Algunos usaran Sprites otros modelos 3D), con perspectiva isométrica/Perpendicular. Basado temáticamente en el inicio del Ragnarok, mitología nórdica visualizada en una realidad paralela. Se pretende crear una experiencia única basada en el modelo básico de Rogué Like bajo una perspectiva distinta ya que no se han encontrado similitudes, con referente a otros títulos. El nombre provisional de este título será "Ragnar-Ok: Life in a Line".

Se propone crear 2 modos de juego 1 base en el cual se generará una dungueon pre hecha y otro modo de juego que cree una dungueon artificial aleatoria. A nivel comparativo e investigación.

Al tener bajo precio será atractivo para el público junto a ser un Rogué Like asequible. Al ser un título cuya duración será alrededor de 30 minutos por nivel como máximo y tener a futuro varios niveles (dungueon) podrá abarcar todo el rango de edades al cual el proyecto se quiere enfocar.

El diseño simple permitirá hacer un "Port" a distintas plataformas a futuro como Android e IOS. A nivel de jugabilidad se pretende agilizar lo suficiente para producir un sistema de combate rápido, así se reducirán la duración de las partidas.

3.1. Definición de objetivos/especificaciones del producto

La idea es crear un nivel 2.5D-3D generado, controlado por una cámara isométrica, así mismo ese nivel estar compuesto de un mínimo de 7 habitaciones con una habitación de Boss como mínimo. Componiendo el lugar por 8 habitaciones en total incluyendo la habitación de inicio.

Objetivos secundarios:

- Crear un menú de inicio, con opciones básicas de gráficos para que sea adaptable a cualquier PC.
- Crear una pantalla de Inicio, Final y Créditos.
- Enemigos con comportamientos personalizados, los cuales cambiaran según su disposición.
- Habitaciones con comportamientos diferentes, así como estéticas distintas.
- Personaje con movimientos ágiles.
- Velocidad de combate media-alta.
- Recolección y uso de consumibles.

Descripción de los objetivos principales que tiene que realizar el producto/servicio (p.ej.: venta de billetes online, reservas de hoteles, alquiler de coches).

Listado detallado de las características, especificaciones y prestaciones principales del producto/servicio (p.ej.: multiplataforma, bilingüe, registro de usuarios, emisión de tarjetas de embarque, sistema de pagos mediante tarjeta de crédito, emisión de facturas).

3.2 Estrategia de marketing

Se propone una venta de marketing digital, el cual se resaltarán las cualidades del producto por distintas redes sociales durante 2 meses aproximadamente. Ya que es un producto digital, resulta innecesario la promoción o marketing del mismo en forma física papel o producto publicitario similar.

Al final del proyecto se crearán cuentas del producto en redes sociales Twitter/Instagram/Facebook promocionando el producto. Cada semana se postearán de 1 a 2 comentarios/historias sobre el producto sobre alrededor de 2-3 meses resaltando imágenes del producto.

Se subirán imágenes a artstation.com y a linkedn.com resaltando el producto en estas redes sociales para su adquisición en la plataforma en donde se suba el producto.

4. Diseño

El entorno de trabajo se ha elegido C-Sharp en Unity usando la plataforma de “Visual Studio Community”. Debido a que es una plataforma versátil a la hora de plantear el código, así como permite la creación de código rápida y segura para la seguida implementación dentro del entorno de la aplicación. Los requisitos técnicos de desarrollo han sido los siguientes:

- Unity Hub
- Unity Editor 2020.3.17.f1
- Visual Studio 2020 Community.

Cabe decir que los siguientes assets están dentro del proyecto, pero su uso y los scripts que usan han sido modificados con el único fin de enriquecer este proyecto. La mayoría pertenecen a Creative Commons o son de libre uso para todos.

Inventario de Assets del proyecto:

- Assets Provenientes de Unity Asset Store de carácter gratuito y libre:
 - o Potions, Coin and Box of Pandora Pack. By Alexander Kotov.
 - o Beautiful Progress Bar. By CYKO.
 - o Dave Pixel “Pirate Coins” models. By DavePixel.
 - o Free Stylized Skybox. By Yuki2022.
 - o Stylize Wood Texture. By LowlyPoly.
 - o Stylized Lava Material. By LowlyPoly.
 - o Stylized Hand-Painted Dungeon. By L2S ARTS.
 - o Japanese Garden Pack. By Waldemarst.
 - o Behavior Bricks. By PadaOne Games.
 - o Zibra Liquids, Free. By Zibra AI.
- Assets Provenientes de otros sitios:
 - o "Kanna Kamui" by xewenet348 in sketchfab.com
 - o "Low-poly Tetsubo" by Mugri & Vatos is licensed under Creative Commons Attribution in sketchfab.com
 - o “Jiggling-Slime” by simulatoralive in opengameart.org

Inventario de Código:

- Procedural Dungeon Scripting 2021 youtube video. By SilverBee
- Procedural toolbox plugin.

4.1. Arquitectura general de la aplicación

Producto desarrollado en lenguaje de programación C-Sharp para la plataforma de Windows, usando como pilar principal los ScriptableObjects como método de serialización y creación de variables utilizables por varios scripts. El producto contiene información básica de la aplicación, así como assets personalizados del producto para el cliente, interacción entre assets-cliente, programable para actualizaciones en la nube a futuro.

4.2. Arquitectura de la información y diagramas de navegación

La arquitectura del producto se ha basado principalmente en los **paquetes de datos** y como estos intervienen dentro de los scripts de forma individual simplificando el uso de recursos por los mismos. Entre los distintos paquetes de datos que se pueden utilizar en Unity, se ha elegido la herramienta de ScriptableObjects, paquetes de datos creados en forma de Asset dentro de la propia aplicación con una base Script, que sirve a Unity Engine para gestionar y crear dichos assets. Estos ScriptableObjects representan conjuntos de paquetes de datos multifunción los cuales pueden ser utilizados por diferentes scripts, en distintos puntos de la aplicación esto permite ahorrar gran cantidad de código referencial dentro de los propios scripts, a su vez debido a la propia arquitectura de los ScriptableObjects cada uno es independiente de cualquier script y permanecen inalterables en su gran mayoría. Estos ScriptableObjects se han creado para que tengan 2 partes:

-**Primera parte no serializable.** Parte que permanece alterable en cualquier momento, la cual es modificada por scripts.

-**Segunda parte serializable.** Parte que permanece inalterable. Esta parte se establece siempre desde el editor de Unity e indica la cantidad o la posición inicial de la variable en cuestión.

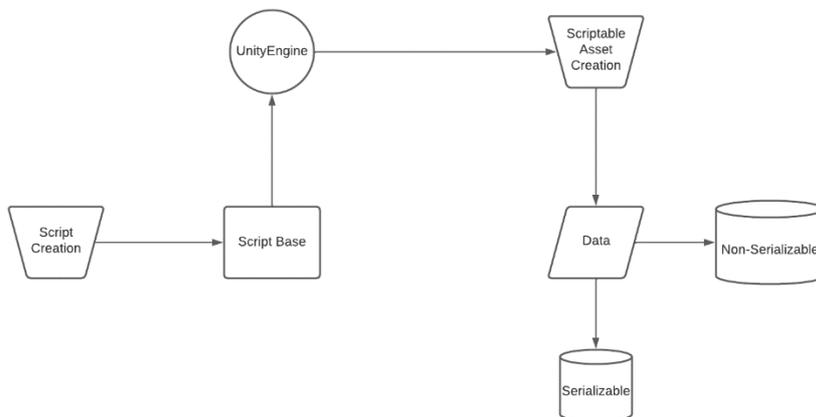


Figura 2: Diagrama creado usando www.lucidchart.com ScriptableObjects.

Usando los ScriptableObjects como pilar, se ha creado una estructura paralela donde scripts con diferentes funciones cogen de datos de forma simultánea. Una ventaja de la cual se ha aprovechado el proyecto es que, si se modifica uno de los datos de uno de estos paquetes de datos, el resto de archivos o scripts que dependen de ese dato se modifican también. Dando como ejemplo la siguiente imagen:

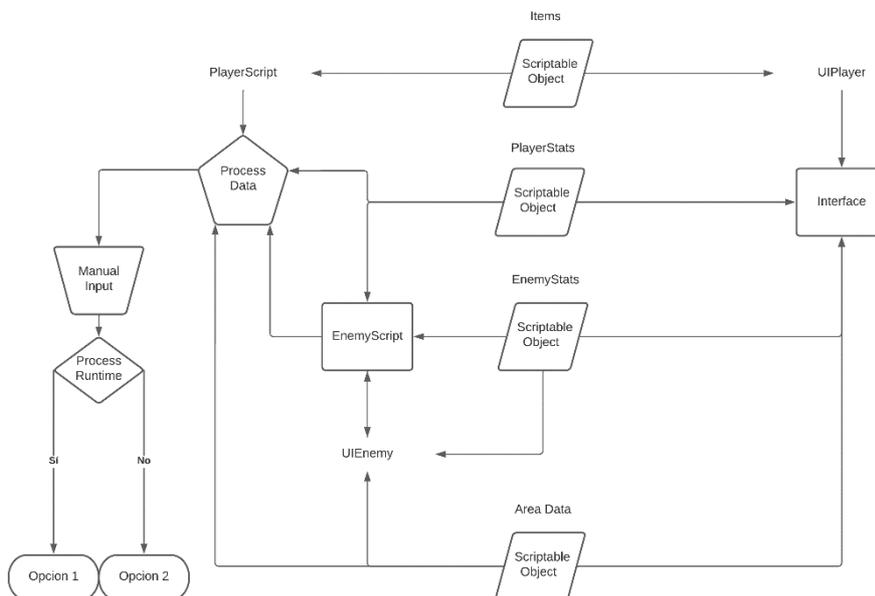


Figura 3: Diagrama creado usando www.lucidchart.com "basic system".

En dicha imagen se puede observar cómo los ScriptableObjects se utilizan por cada uno de los componentes de manera individual y no interactúan entre sí.

4.3. Diseño gráfico e interfaces

Modelos propuestos para el proyecto. Se han propuesto 2 tipos de modelo basados en 2 paletas de colores completamente distintas.

4.3.1. Estilos

La primera propuesta de estilo es una de colores en blanco y negro con tonos apagados. Resaltando el hecho de que lo que pretende impactar de la estética es el cielo que se cae, la primera paleta de colores propuesta para la interfaz de usuario es una oscura y blanca.



Figura 4: Paleta de colores básica de tonos blanco a negro.

O una basada en 3 colores, basados en colores vivos en el cielo.



Figura 5: Paleta de colores obtenida de la página color Palletes de Adobe.

Para el título en cuestión, se ha buscado una tipografía gratuita de carácter libre. “Valorant” de dafont.com

La cual se usará para el título principal y el menú de opciones.

Prototipado general Opción 1 Menú Principal:

-Optando por un diseño clásico y un menú simplista.

-Fuente clara y visible.



Figura 6: Hipotético menú para el juego con título y botones. Creada usando Adobe Photoshop e Illustrator.

Al final se ha escogido por usar la interfaz básica del juego modificada con botones creados propios. Debido a que se quiere que el juego sea lo mas ligero posible para este proyecto, por lo que será lo único añadido al menu.

4.3.2. Usabilidad /UX

Para la interfaz de usuario se han creado 2 opciones dependiendo de la paleta grafica que se escoja:

Primera Opción Blanco y Negro:



Figura 7: Hipotético User Interface. Creada usando Adobe Photoshop e Illustrator e imágenes in Engine.

Donde arriba siempre estará la vida y la estamina junto con una foto del personaje, mientras que abajo a la derecha los consumibles (a futuro). Los enemigos se ha decidido que tengan una barra de vida que se active por proximidad.

Segunda Opción Paleta de colores:

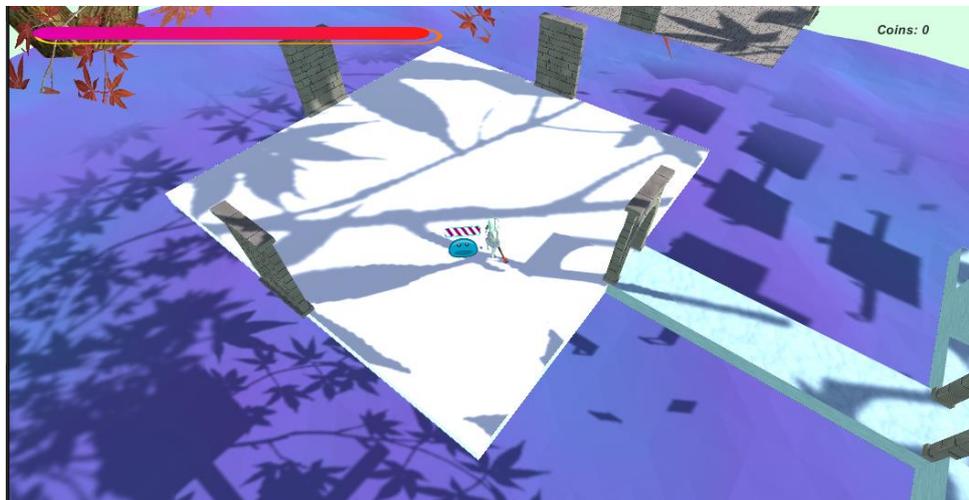


Figura 8: Hipotético User Interface. Creada e implementada temporalmente en Unity para testeo, imagen In Engine.

Opción más visible para el jugador, ya que resalta más a la vista para el mismo, utilizando los assets del paquete Beautiful Progress Bar. Usando una situación similar, pero limitando la iconografía a más simple aún. Dejando la estamina como algo invisible.

Aun no se han escogido ninguna de las 2 opciones dadas sin embargo se estima que se hará una mezcla de los 2 en el producto final quedando una estética de bordes negros cuadrados con tonos que resaltan.

Al no haber incluido mecánicas complejas al juego base se mantendrá solo la barra de vida principal del jugador y la barra de los enemigos.

4.4. Lenguajes de programación y APIs utilizados

Actualmente se está utilizando el lenguaje nativo de unity C Sharp, aunque Visual Studio integrado permite programar en distintos lenguajes de programación, se ha decidido utilizar este en específico debido a que últimamente junto a C++ son los lenguajes de programación en auge para el desarrollo de videojuegos.

Enumerando los distintos tipos de lenguajes y asociándolos a una plataforma se ha decidido la plataforma de Unity por la versatilidad que esta ofrece y los distintos productos que se encuentran en

creative commons para todo el público, permitiendo una mayor personalización de los proyectos a desarrollar.

En comparativa con otras plataformas, se debe hablar de Unreal Engine, esta posee una gran cantidad de assets para libre consumo, además de tener un manejo de assets bastante diferente a Unity, aunque intuitivo. Por otro lado, utiliza un lenguaje de programación bastante diferente C++, aunque se ha visto que se puede implementar C Sharp a través de plugin, el uso de recursos de los equipos informáticos en general ha resultado mayor, por lo que se ha optado en descartar el uso de Unreal Engine para la creación de este producto.

Otras plataformas que también se han considerado para el proyecto: GameMakerStudio, Godoot.

Estas plataformas han sido descartadas por 2 motivos.

-Utilizan un lenguaje nuevo o bastante distinto a C Sharp. Sobre todo, Godoot que utiliza una serie de assets y referencias bastante distintas.

-El Carácter estético que podrían ofrecer no era lo que se esperaba para este producto.

Por consiguiente, el producto final se ha decidido crear y aplicar en Unity usando Visual Studio 2020

5. Implementación

La implementación de este producto se hará por parte del sistema que ofrece unity de crear un ejecutable junto un paquete básico de datos. Por consiguiente, no habrá un paquete de instalación o en su caso una descarga salvo la inicial que incluirá todo. Al ser un producto básico no requiere de una gran cantidad de almacenamiento para hacerlo funcionar.

5.1. Instrucciones de instalación

- Descargar el contenido digital vía descarga directa.
- Alojar el contenido en la carpeta deseada.
- Ejecutar.

6. Demostración

6.1. Instrucciones de uso

- Se inicia el juego.
- Se presiona Espacio para llegar al menú principal.
- Se presiona Start.
- Movimientos ASDW y letra F o Clic izquierdo del mouse.
- Letra P para Pausar el juego.

6.2. Prototipos

Prototipos creados a lo largo del proceso de desarrollo.

6.2.1. Prototipos Lo-Fi

- Sketches

Bocetos Conceptuales:

-Personajes:



Figura 9: Boceto Conceptual Personaje Principal.

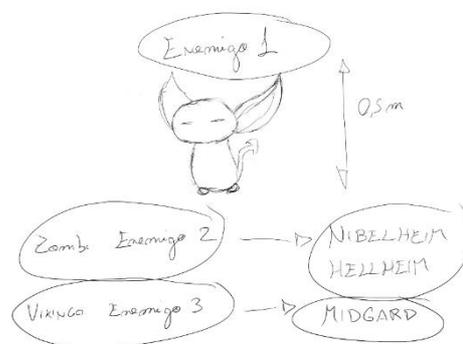


Figura 10: Boceto Conceptual Enemigo Básico.

-Visión General:

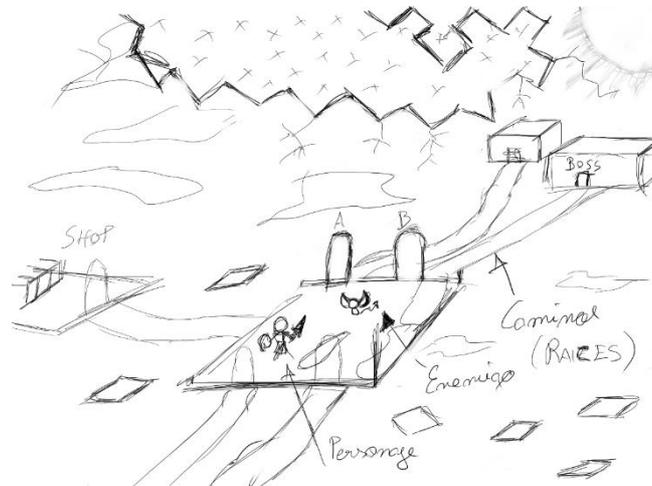


Figura 11: Boceto Conceptual Visión del Mundo.

6.2.2. Prototipos Hi-Fi

- MockUp

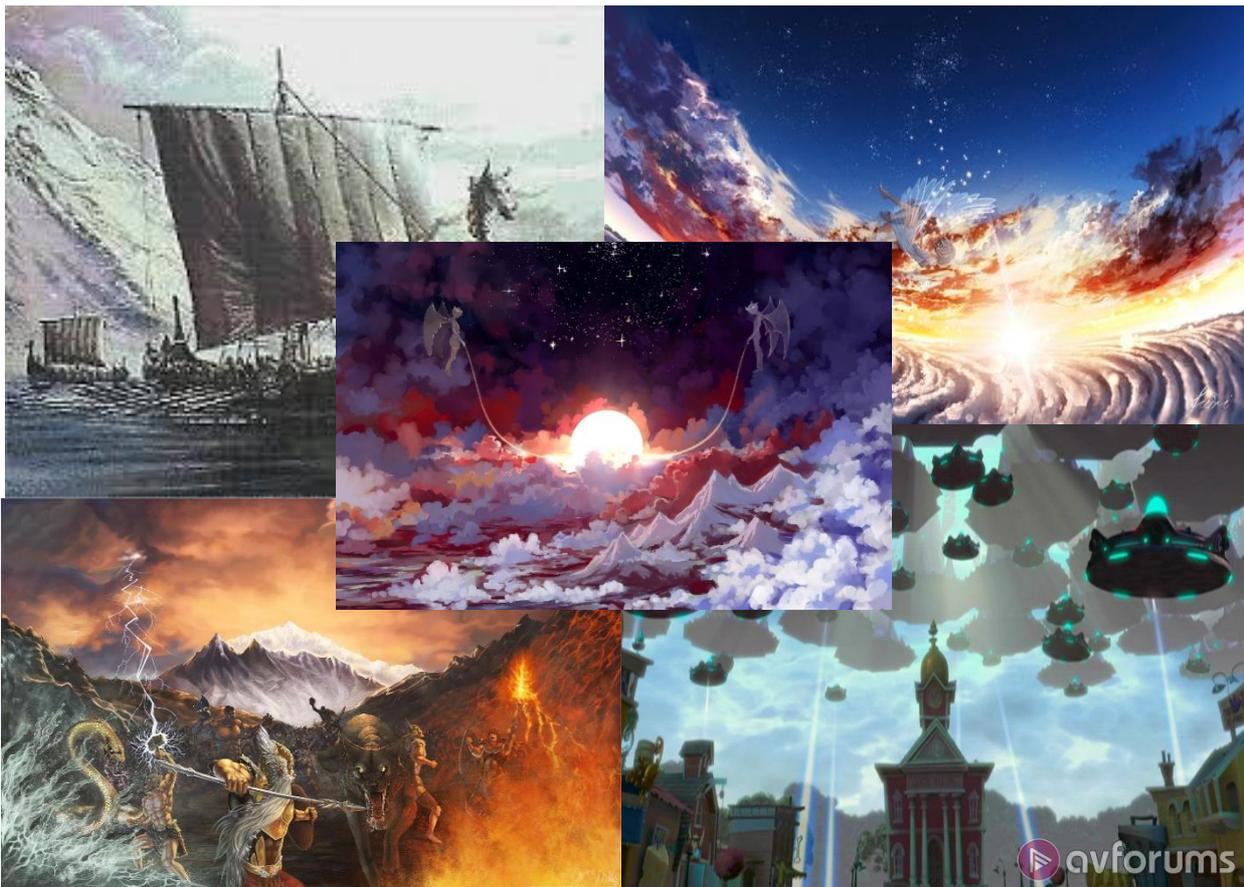


Figura 12: MockUp Creado a partir de imágenes de varias páginas de internet.

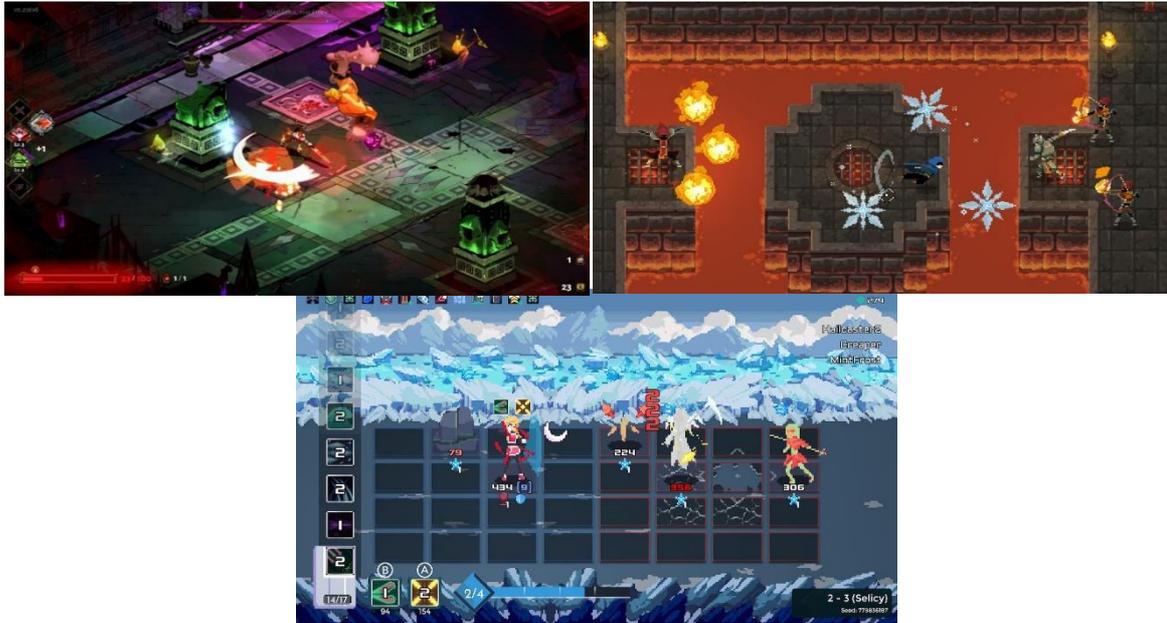


Figura 13: MockUp 2 Estilos de juego similares, dungeon crawler y otros.

6.3. Ejemplos de uso del producto (o guía de usuario)

- Se inicia el juego.
- Se presiona Espacio para llegar al menú principal.
- Se presiona Start, Options, Exit.
- Iniciar el juego
 - +Movimientos ASDW y letra F o Clic izquierdo del mouse.
 - +Letra P para Pausar el juego.
- Si el jugador muere pantalla de Game Over.

7. Conclusiones y líneas de futuro

7.1. Conclusiones

Como conclusión personal, ha sido una gran experiencia el crear un juego tan metódico como puede ser un juego generado de manera procedural. Se es consciente de que cabe mucho margen de ampliación para el proyecto debido a las limitaciones técnicas en su gran mayoría. Por otro lado, personalmente se ha dado la ocasión a experimentar con herramientas muy diversas como es el uso de la NavMesh en Run time, la cual ha llevado tiempo investigar (y muchos errores durante el camino). Como conclusión oficial, el proyecto ha cumplido su cometido inicial y es el de crear una experiencia algo única, también permite su comparación a nivel técnico a través del editor de Unity.

Lecciones aprendidas:

-El uso y dificultad de coordinar varios grupos de variables en un entorno cambiante.

Ya que modificar continuamente el entorno durante el propio juego nunca ha sido parte principal del propio proyecto, pero se han dado casos durante el desarrollo del mismo que han dificultado el generar todo y al superar dichas dificultades ha podido cambiar mi perspectiva a la hora de crear este trabajo.

Por desgracia siendo objetivo, inicialmente este artículo ofrecía una comparativa, cosa que no se ha hecho. Debido a que se vio innecesaria la creación manual de un escenario debido a que mismamente el mismo código creado, permitía hacer lo que se pretendía de modo manual con menos dificultad y tiempo. No se piensa que esto sea algo negativo, ya que mismamente se ha logrado crear un producto que funciona. Por otro lado al intentar que el producto fuese lo menos pesado posible y a su vez que sea funcional para este proyecto la calidad gráfica se ha reducido considerablemente.

A nivel de planificación el proyecto ha seguido el plan de proyecto determinantemente. No han ocurrido retrasos a nivel de entrega, ni fallos no previstos.

7.2. Líneas de futuro

A Futuro si se continua este proyecto es posible crear algo aún mayor. Las bases de código son buenas debido a que mismamente, este es muy permisivo a la hora de introducir código nuevo o código de otras fuentes. El uso de ScriptableObjects permite el crear paquetes de datos nuevos en cualquier momento, además de que se ha pensado a futuro con esto mismo y hay muchas variables actualmente que no tienen un uso ahora por que se tiene visión a futuro del proyecto. Algo en lo que no se tuvo en cuenta es en el uso de sonidos cosa que no se ha aplicado a ninguna parte del proyecto.

Considerando cosas que se podrían añadir a futuro: Escenarios, decoraciones, tiendas, magias, sistemas de combate más complejos, un buen menú, enemigos nuevos, más bosses, más mecánicas...

Personalmente creo que, aunque han sido varios meses bastante ocupados, el trabajo como tal puede ampliarse hasta crear un producto bastante accesible y muy entretenido.

Bibliografía

Unity Script Manual: <https://docs.unity3d.com/Manual/ScriptingSection.html>