Universitat Oberta de Catalunya

uoc.edu

Las Historias de Anvein:
El poder de la Nominación

Andoni Bisbal Castaño
Máster en diseño y desarrollo de videojuegos
Videojuego Completo

Joan Arnedo Moreno
Helio Tejedor Navarro

07/06/2020

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Las Historias de Anvein: El poder de la Nominación*
Nombre del autor:	*Andoni Bisbal Castaño*
Nombre del consultor/a:	*Helio Tejedor Navarro*
Nombre del PRA:	*Joan Arnedo Moreno*
Fecha de entrega (mm/aaaa):	06/2020
Titulación::	*Máster en diseño y desarrollo de videojuegos*
Área del Trabajo Final:	*Trabajo Final de Máster*
Idioma del trabajo:	*Castellano*
Palabras clave	*Máximo 3 palabras clave, validadas por el director del trabajo (dadas por los estudiantes o en base a listados, tesauros, etc.)*

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

La finalidad del trabajo era realizar un juego completo, es decir, un juego que se pudiera comercializar. Para ello, se ha realizado un juego de plataformas 2D con una mecánica de cambio de gravedad.

El juego incluye todas las pantallas de un juego completo y varios niveles de plataformas.

Se ha usado la metodología Scrum para realizar el trabajo en Sprints de dos semanas durante el semestre, gracias a ello se han obtenido unos buenos resultados en el trabajo. Se ha utilizado Unity como motor gráfico.

El juego se ha probado en varios jugadores, dando feedback y pudiendo añadir funcionalidades no contempladas al inicio del desarrollo.

Como conclusión, ha sido un buen trabajo para ver todo el desarrollo que hay detrás de un juego y la cantidad de gente que puede llegar a participar en el.

Abstract (in English, 250 words or less):

The purpose of the work was to make a complete game, that is a game that could be commercialised. For this, a 2D platform game has been made with a gravity change mechanic.

The game includes all the screens of a complete game and differents platforms' levels.

To create this world's game we applied the Scrum methodology, with two-week sprints during the semester.

Hence we obtained good results. For this work we used Unity as the graphics engine.

The game has been tested in several players, giving feedback and being able to add unforeseen functionalities at the beginning of development.

In conclusion, it has been a good job to realize all the development behind a videogame and the number of people that could participate in it.

Índice

1. Introducción

1.1 Contexto y justificación del Trabajo

Se ha realizado un juego completo para poder ver todos los puntos a tener en cuenta al crear un videojuego. Después de lo aprendido durante dos años, es la forma de poner en practica todas las herramientas aprendidas y unirlas en un producto final.

1.2 Objetivos del Trabajo

Los objetivos del trabajo han sido los siguientes:

1. Poner las bases de una idea para videojuego.
2. Escoger las herramientas de desarrollo para el tiempo limitado del mismo.
3. Plantear un roadmap del desarrollo.
4. Implementar las ideas.
5. Desarrollar un videojuego comercializable.

1.3 Enfoque y método seguido

Se ha decidido desarrollar un producto nuevo, basado en juegos de plataforma 2D. Hemos escogido este enfoque para desarrollar un nuevo tipo de juegos 2D cambiando el salto por la habilidad de cambiar la gravedad. Con ello se consiguen nuevas mecánicas y nuevos enfoques para los juegos 2D.

Se ha seguido la metodología Scrum, para poner a prueba los nuevos avances en esta metodología y comprobar que también sirve para el desarrollo individual.

1.4 Planificación del Trabajo

Se ha realizado el trabajo con Unity como herramienta de trabajo.
La planificación del trabajo se ha realizado con sprints de dos semanas con diversos objetivos separados.
Objetivos de la planificación:

1. Movimientos del juegador principal y primer nivel
2. Movimientos de los montruos básicos, los bandidos, gestión de elementos y recursos
3. Diseño de pantallas del Menú y genericas del juego y gestión músical
4. Movimiento de los tres boses
5. Diseño de los niveles de cada bloque. Tres niveles por bloque. Tres bloques
6. Gestión de guardado de partida
7. Corrección de errores y dos funcionalidades nuevas para el jugador

1.5 Breve sumario de productos obtenidos

Los productos externos para el desarrollo han sido:

1. Música ambiente general: Laia Guilanyà Jané
2. Música Anvein: Laia Guilanyà Jané
3. Música Castillos y Bosses: Laia Guilanyà Jané
4. Recursos de efectos sonoros: freesound.org
5. Recursos visuales: Asset Store Unity

1.6 Breve descripción de los otros capítulos de la memoria

1. Estado del Arte: Revisión del género del juego y de la tecnología usada
2. Definición del juego: Mostrar el GDD (Game Design Document) generado al inicio del desarrollo, con las mejoras o cambios pertinentes
3. Diseño técnico: Explicación del entorno usado y de la creación del juego, su desarrollo y estructura
4. Diseño de niveles: Explicación de los niveles
5. Manual de usuario: Manual para el uso del producto
6. Pruebas de usuario: Explicación de las pruebas y feedback de varios usuarios
7. Conclusiones: Conclusiones del trabajo realizado y puntos de mejora.

2. Estado del arte

2.1 Revisión del género del videojuego

El videojuego creado es del género de plataformas 2D con scroll horizontal. Este tipo de género era muy típico en las primeras videoconsolas que salieron al mercado, con clásicos como el Super Mario Bros, Donkey Kong, etc. Actualmente, este tipo de juegos, se realizan sobretodo para dispositivos mobiles o tablets.

Antes de empezar, probé varios juegos de este estilo, como Swordigo, Dan On The Man, Total Party Kill (muy recomendado). Como patrón era que todos tenían el movimiento horizontal y el salto, después las mecánicas suelen ser muy similares, variando los ataques y los niveles. Cierto es, que Total Party Kill tiene un sistema de puzzles muy divertido e innovador, pero de todos modos, quería introducir un nuevo estilo de mecánicas, jugando con la gravedad, y allí fue donde nació la idea del videojuego. De esta forma, no perdemos este gran género clásico y le damos innovación al mismo tiempo, para poder dar más alicientes a los jugadores.

2.2 Revisión de la técnologia

Actualmente el motor más usado para este tipo de videojuegos suele ser Unity, debido a su gran comunidad y que este motor empezó para videojuegos 2D. Tiene una buena estructura para su desarrollo.

También existen otros motores, centrados en lenguaje JavaScript, que se suelen usar para hacer juegos para versión Web o Android.

El motor suele ser más simple que un motor 3D, incluso se podría programar desde 0 con C/C++ o con la librería de Raylib, por ejemplo.

También destacar, que en este tipo de videojuegos se trabaja mucho con los Sprites, siendo la fuente principal del arte visual del juego, y creando animaciones, haciendo cambios de estos.

Por otro lado, los niveles se suelen diseñar a través de una paleta de tilesets, que son imágenes troceadas en cuadrados iguales, donde cada cuadrado contiene algún bloque para crear el mundo, de esta manera, asignando cada punto de la imagen a un indice, se pueden crear mapas de forma rápida y variada.

Actualmente, es muy habitual ver este tipo de juegos, con el arte de Pixel Art, que siempre da un toque más retro.

3. Definición del juego

En este apartado mostraremos el Game Design Document realizado para el desarrollo de este videojuego.

3.1 Información General

- **Título:** Las Historias de Anvein: El Poder de la Nominación
- **Género:** Acción, Acción-Aventura 2D, Plataformas
- **Plataforma de destino:** Desktop
- **Motor:** Unity

3.2 Idea del Juego

- **Descripción del juego**

En un mundo controlado por poderosos magos y oscuros secretos, ha nacido un héroe sin precedentes, Anvein.

Encarnando a Anvein, viajarás a través del mundo Nominador, con los poderes que te han concedido saber el nombre correcto de las cosas o comúnmente llamado, Nominación.
Anvein ha conseguido controlar la gravedad y ahora va en busca del verdadero nombre de las fuerzas naturales del universo.

Recorre el planeta Nominador con Anvein, a través de un mundo 2D con grandes aventuras y mazmorras, consigue derrotar los magos poseedores del poder de la Nominación y aumenta tu propio poder.

- **Subgénero y referencias**

  Juego de acción y aventuras en un mundo 2D de plataformas y puzzles para superar los niveles.
  Como referencia principal tenemos al clásico Super Mario Bros. Las referencias que han ido puliendo las mecánicas y dinámicas del juego han sido Swordigo y Dan of the Man. Con ellos he empezado a implementar la idea narrativa a un estilo de juego 2D que sea dinámico, con una historia completa, con posibilidad a futuras secuelas.

- **Tipo de interacción juego-jugador**

  La interacción juego-jugador será a través del teclado, con una camera 2D que se moverá con el jugador, mostrando siempre la misma distancia a todos los puntos.
  La intención de la interacción, es introducir al jugador a una serie de retos mecánicos y puzzle mientras descubre la historia del personaje y del mundo, a través de textos explicativos y interacción con terceros.
  No se busca interacción para conseguir más objetos o ser el mejor en un ranking, sino un juego para disfrutar de su historia y de su jugabilidad, volver a los primeros juegos 2D de plataformas con historias sencillas y buenas dinámicas, haciendo disfrutar el jugador y tener retos consigo mismo. La estética pixel art, aportará ese sentimiento nostálgico del 2D.

  Los controles serán los siguientes:
  - **Ratón:**
    - Control de los botones de los menús y compra de objetos en tienda
    - Botón izquierdo control de ataque básico jugador
    - Botón derecho ataque fuerte jugador
  - **Teclado:**
    - A-D: Movimiento horizontal del jugador
    - Space: Cambio del sentido de la gravedad

- W: Ataque de repulsión
- S: Ataque de atracción
- 1: Power-up Eléctrico
- 2: Power-up Fuego
- 3: Power-up Fuego
- E: Entrar tienda/interacción
- I: Inventario
- ESC: Pausa

- **Plataformas de destino**

  La plataforma principal sera el desktop de un ordenador con windows. De todas formas, puede tener futuras versiones para Android/IOS y Nintendo Switch. La principal elección para la plataforma desktop, es concentrar el juego en su historia y sus mecánicas dejando de lado las tendencias actuales de plataformas Android/IOS de centrar el juego en conseguir coins, u objetos similares, para aumentar tu potencial en el juego o simplemente modificar la vestimenta.
  Se escoge para dar sentido al juego, a la programación del mismo y el disfrute de una historia.

3.3 Conceptualización

- **Historia, ambientación y/o trama**

  Anvein es un joven nominador que ha aprendido el poder de la nominación a través de la lectura de un viejo libro. Gracias a ello, a descubierto como dominar el poder de la gravedad a su antojo.
  En el mismo libro, se menciona que el poder de la nominación se puede llevar a casi todos los elementos naturales o fuerzas existentes en el mundo Nominador, pero no da más pistas.
  Anvein decide recorrer el mundo en busca de más pistas sobre este fantástico poder y convertirse en un gran maestro Nominador.

Al principio de su aventura descubrirá que existen dos maestros que mantienen en secreto los poderes nominadores del electromagnetismo y de la termodinámica, por lo que decidirá ir conseguir estos poderes.
Se ambientará en un mundo imaginario de estética medieval, donde los grandes deseos de conocer llevarán a Anvein a descubrir su propio planeta.

Se realiza la trama en torno a estas tres fuerzas físicas debido a mi carrera universitaria, grado en física, como un pequeño homenaje a ello y también para abrir un nuevo mundo de poderes en los juegos de de ambientación fantástica.

La forma de obtener los poderes, la nominación, proviene de una saga de libros de fantasía (Crónicas del asesino de Reyes), y me parece una bonita forma de introducir magia en los distintos mundos, el poder de saber el verdadero nombre de las cosas. Y dar un pequeño homenaje a mi libro favorito, El nombre del Viento.

- **Definición de los personajes/elementos**

  - **Personajes Principales**

    - **Anvein:** Personaje principal de la historia, un joven divertido y con ganas de descubrir el mundo. Su pasión por la lectura lo ha llevado a sus aventuras.
    - **Maestro Tesla:** Maestro que controla el poder del electromagnetismo, primer maestro que se encontrará el héroe.
    - **Maestro Entrópico:** Maestro que controla el poder de la termodinámica, segundo maestro que se encontrará el héroe.
    - **Maestro Decano:** Maestro final del juego, controla los 3 poderes que ha conseguido Anvein, un malvado maestro que quiere dominar el planeta. Los dos anteriores eran sus discípulos.

- **Personajes Secundarios**
  - **Bandidos:** Humanos que están por el camino infundiendo miedo al pueblo para beneficio del gran Decano. Son hábiles con la espada, pero no tienen poderes.
  - **Monstruos:** Monstruos del propio planeta que atacan a los humanos cuando los ven, suelen estar por el camino. Menos poderosos que los bandidos.

- **Elementos**
  - **Monedas:** Monedas como recompensa en sitios del nivel, que darán acceso a comprar comida para recuperar vida. Tres tipos, Oro, Playa y cobre
  - **Comida:** Recuperarás un porcentaje del total de la vida. Solo se consigue con dinero, en pequeños establecimientos dentro de los niveles. Tres tipos, Manzana, Jamón y Cerveza
  - **Tienda:** Pequeña tienda con alimentos
  - **Trampas:** En todos los niveles habrán trampas de 4 tipos. Aspas gigantes que giran, un circulo en forma de sierra, una bomba que se mueve en vertical y lanzadores de flechas
  - **Inventario:** Sitio donde podremos consumir la comida comprada
  - **Mundo:** El mundo constará de dos partes, niveles exteriores y nivel final de cada zona que será el interior de un castillo. 3 niveles por zona, 3 zonas en todo el juego. Nivel 0, introducción del juego y movimientos.

- **Interacción entre los actores del juego**

  La interacción de los personajes constará en su interacción con ataque-defensa.
  Los monstruos te perseguirán y si te tocan te harán daño, movimiento simple e IA de persecución. No tienen mucha vida.

Los bandidos, tendrán algo más de vida y te atacarán con la espada.
Los maestros principales, subirán el nivel de dificultad, con más variación de ataque y mucha vida.
Las monedas, corazones o comida se activarán al cruzar los elementos.
La tienda será una escena aparte, donde podemos intercambiar monedas por comida, que nos subirá la vida al momento.
Las flechas nos quitarán un 20% de la vida total.

- **Objetivos planteados al jugador**

Superar los distintos niveles del juego y conseguir todos los poderes de la nominación.
Derrotar a los maestros como enemigos principales de la historia.
Se busca gran habilidad con los juegos de plataformas y pensar como resolver ciertos puzzles del juego. Estrategias de ataque y defensa en los combates.

- **Concept Art**

Los distintos sprites usados, se conseguirán de la assetStore de Unity. A continuación imágenes de los personajes y el estilo de niveles.

- Anvein:

- Bandidos:

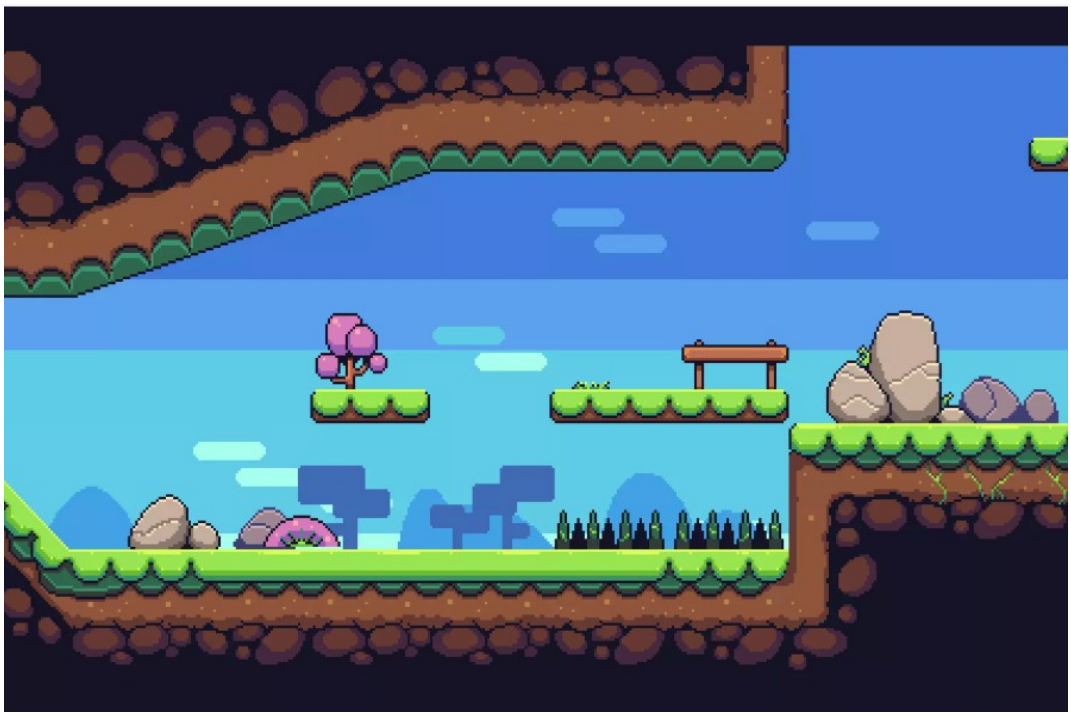
- Maestros Tesla y Entropico (Mismo estilo de personaje):

- Monstruos:

- Maestro Decano:

- Objetos Varios:

- Mundo exterior:



- Castillo:

3.4 Desarrollo y Roadmap

- **Evaluación de engines y kits de desarrollo**

  Para realizar el proyecto usaremos el Engine de Unity con todos sus kits de desarrollo y los assets de su pagina web, tal y como hemos mostrado anteriormente.
  La decisión de usar Unity se basa en dos términos, el primero es que ya estoy familiarizado con él y eso ayudará en los tiempos de desarrollo, evitando el tiempo de aprendizaje de un Engine nuevo. Por otro lado, Unity permite una exportación rápida y sencilla para su ejecución en Desktop, nuestro principal sitio de producción, y por otro lado también sería fácil exportarlo a formato móvil, el otro lugar donde podemos darle salida al videojuego en un futuro.
  Al ser 2D, Unity tiene un gran desarrollo del motor en 2D, y el estilo gráfico también es el correcto con Unity, ya que no necesitamos un gran sistema de renderización. También nos aprovechamos de la gran comunidad que tiene Unity para los distintos problemas que puedan surgir en un futuro.

  En conclusión, para este proyecto, y por el tiempo limitado de producción, el mejor motor es Unity. De todos modos, he estado revisando otros motores para sacarlo a nivel web, que son orientados a JavaScript, y que puede ser un buen trabajo de futuro traspasarlo a uno de estos engines.

- **Planificación de objetivos y Cuantificación de tiempo y recursos por objetivo**

  La planificación de objetivos la vamos a dividir en distintos bloques, para así poder trabajar mejor durante el curso. De cada bloque general sacaremos pequeñas tareas que podremos cuantificar en tiempo.
  Utilizare la metodología Scrum de trabajo con sprints de 2 semanas.

3.5 Bloques

- **Pantallas No GamePlay → total: 43,5 horas**
  - Pantalla Logo:
    - ➔ Realización del Logo → 1 hora
    - ➔ Realización de la Pantalla sin interacción → 1 hora
  - Pantalla título:
    - ➔ Realización de la Pantalla con interacción “botón start” → 1 hora
  - Pantalla del Menú:
    - ➔ Realización de la Pantalla con 3 botones → 2 horas

      (Pantalla con los botones de iniciar/seguir partida, ir a pantalla de selección de niveles, ir a pantalla de opciones, ir a créditos)
  - Diseño de pantalla historia inicial de algunos niveles
    - ➔ Clip historia inicial del juego → 5 horas
    - ➔ MiniClips historias en cada nivel → 8 horas
  - Diseño de pantalla de historia final de algunos niveles
    - ➔ Final de cada Mago → 10 horas
    - ➔ Final de cada nivel → 6 horas
  - Pantalla GameOver:
    - ➔ Realización de la pantalla con interacción “botón volver título” o “repetir nivel” → 1,5 horas
  - Pantalla Opciones:
    - ➔ Realización de la pantalla con botones para silenciar música y fx sounds → 2,5 horas
  - Pantalla Elección niveles:
    - ➔ Realización de la pantalla con los niveles ya realizados → 3 horas
  - Pantalla Pause en Gameplay:
    - ➔ Pantalla con 3 botones: → 1,5 horas

      (botón ir al menú, botón reanudar, botón reiniciar)
  - Pantalla de Créditos:

- ➔ Pantalla con los créditos del juego y “botón volver a menú/título → 1 horas

- **Niveles de juego → total de 40 horas**
  - Bloque Inicial:
    - ➔ 1 nivel inicial a modo de tutorial de las mecánicas del juego, con introducción visual a la historia a modo de Clip. → 10 horas
  - Bloque Mago Tesla:
    - ➔ 4 niveles, 3 niveles exterior y un nivel modo mazmorra → 10 horas
      - Diseño de niveles
      - Diseño puzzles
  - Bloque Mago Entropico:
    - ➔ 4 niveles, 3 niveles exterior y un nivel modo mazmorra → 10 horas
      - Diseño de niveles
      - Diseño puzzles
  - Bloque Mago Decano:
    - ➔ 4 niveles, 3 niveles exterior y un nivel modo mazmorra → 10 horas
      - Diseño de niveles
      - Diseño puzzles

- **Movimientos Jugador Principal → total de 30,5**
  - Movimientos básicos:
    - ➔ Control horizontal del personaje y movimiento de cambió del sentido de la gravedad → 5 horas
  - Movimientos de ataque:
    - ➔ Ataque con la espada simple y avanzado → 3 horas
    - ➔ Movimiento de atracción del enemigo → 3 horas
    - ➔ Movimiento de repulsión del enemigo → 3 horas
  - Power-Ups:

        - ➔ Movimientos de ataque con electricidad → 3,5 horas
        - ➔ Movimiento de ataque con fuego → 3,5 horas
        - ➔ Movimientos de ataque con hielo → 3,5 horas
    - Movimientos defensivos:
        - ➔ Bloqueo de ataque → 2 horas
    - Movimientos de herido y muerto:
        - ➔ Movimiento de herido → 2 horas
        - ➔ Movimiento de muerto → 2 horas

- **Movimientos Magos finales → total de 36 horas**
    - Movimiento IA:
        - ➔ Movimientos del personaje → 8 horas
    - Movimientos ataques:
        - ➔ Movimientos de ataque simples → 6 horas
        - ➔ Movimientos de ataque complejos → 10 horas
    - Movimientos defensivos:
        - ➔ Movimientos defensivos, definición y ejecución → 12 horas
- **Movimientos Monstruos → total de 5 horas**
    - Movimiento IA:
        - ➔ Movimientos de los personajes → 3 horas
        - ➔ Colliders de colisión, estilo setas Super Mario → 2 horas
- **Movimientos Bandidos → total de 7 horas**
    - Movimiento IA:
        - ➔ Movimientos de los personajes → 3 horas
    - Movimientos ataques:
        - ➔ Movimientos simples de ataque → 4 horas
- **Gestión de elementos y recursos → total de 18 horas**
    - Gestión de la vida del personaje → 1 hora
    - Gestión de la vida de los enemigos: → 1 hora
    - Obtención de corazones y comida: → 1 hora
    - Obtención de monedas: → 1 hora

- Gestión de flechas y trampas: → 4 horas
- Gestión de la tienda interna del juego. Con dinero virtual para conseguir comida → 10 horas

- **Gestión de Guardar Partida → total de 25 horas**
  - Guardado de partida según nivel → 10 horas
  - Desbloqueo de selección de niveles → 5 horas
  - Continuar desde el último nivel no conseguido → 5 horas
  - Guardado de inventario → 5 horas

El total de horas a la que suben las tareas son 204 horas y le sumaremos un 15% del total para tener margen de error en la estimación, lo que nos dará un total de 234,6, lo que redondeamos a 235 horas de trabajo.

Contando que empezaremos el día 9 de marzo el desarrollo y con sprints de 2 semanas, lo que tenemos un total de 6 sprints.

Por lo tanto tendremos un tiempo estimado de desarrollo de 40 horas por sprint, lo que son 20 horas semanales. Se entiende que en la estimación se ha tenido en cuenta las gestión de pruebas y errores del programa.

- **Conclusiones al finalizar el juego**

  Después de realizar el juego, se han realizado cambios en el numero de niveles, se han añadido trampas y 2 funcionalidades nuevas, que no entraban en esta primera toma de requerimientos realizada. Pero he encontrado interesante ver las funcionalidades planteadas al inicio y los cambios realizados. Los Sprints realizados durante el semestre, han echo que se puedan realizar estos cambios para tener un producto mejor acabado. Por lo que concluimos que al seguir el roadmap establecido, hemos podido detectar errores y modificarlos, también hemos podido realizar un estudio para nuevas funcionalidades y dejar un juego equilibrado y completo.

A continuación una tabla con los cambios entre el roadmap inicial y el juego final:

Requerimientos Iniciales	Juego Final
4 niveles por bloque	3 niveles por bloque
Solo flechas	4 trampas distintas
Vídeos cortos inicio y final niveles	-
-	Inventario
-	Control de ataques mágicos (mana)

4. Diseño técnico

A continuación explicaremos el diseño técnico del proyecto en diferentes apartados, tanto a nivel de herramientas como de diseño de la solución para los requerimientos previos mencionados.

4.1 Entorno de desarrollo

Para desarrollar el juego, se ha utilizado un ordenador con sistema operativo windows 10.

El desarrollo del juego se ha realizado mediante el motor gráfico Unity, en la versión 2019.3.8f1.

Se ha escogido el motor Unity por dos grandes motivos, el primero ha sido el tiempo de desarrollo y la familiarización con el motor, y el segundo, porqué es un buen motor para desarrollo en 2D, como el juego planteado. También comentaremos más brevemente, otros aspectos que nos han llevado a esta elección.

El primer aspecto importante ha sido el tiempo de desarrollo, alrededor de 3 meses, lo cual ponía un tiempo limite, por lo que lo mejor era escoger un entorno en el cual me sintiera cómodo y tuviera rapidez para desplegar las diferentes funcionalidades planteadas. Mi experiencia en

Unity me ayudado para mejorar los tiempos y tener un mejor producto final. Mi gran ilusión es haber podido realizar mi propio motor para el desarrollo de este videojuego, pero esto lo dejo para el futuro.
El segundo punto, ha sido el buen rendimiento de Unity con juegos 2D, debido que el otro motor que tenia nociones, UnrealEngine4, esta más establecido para buenos gráficos y un formato de desarrollo 3D. Este motivo ha echo que me centrara en Unity y su facilidad de componentes 2D. La versión también se ha escogido para hacer uso de los tilemaps que se han introducido al motor para la creación de niveles.

Otros aspectos a tener en cuenta, han sido la experiencia en Unity tanto en el máster como en cursos echos anteriormente, la gran comunidad y fácil búsqueda de problemas, errores o soluciones algunos requerimientos que podemos encontrar en Internet y el soporte de la propia pagina de Unity. También, su Asset Store, que para gente como yo, que no tenemos mucho arte para dibujar, nos ayuda a poder desarrollar juegos con diseños de otra gente, tanto de forma gratuita como pagando.

Como he mencionado, mi ilusión es montar mi propio motor para este tipo de desarrollos, y creo que Unity es una buena referencia para poder organizar este trabajo, por lo que este desarrollo también ha servido a este propósito.

4.2 Requerimientos técnicos

A continuación mostraremos un listado de los requerimientos técnicos usados para el desarrollo de este videojuego y después para poder ejecutarlo.

Requerimientos técnicos del desarrollo:

1. Sistema operativo: Windows 10

2. Procesador: Intel core i5 7th generación
3. Tarjeta gráfica: Nvidia GeForce GTX 1050

Requerimientos técnicos para jugarlo:

1. Sistema operativo: Windows 7, 8 y 10, linux y MacOS
2. Procesador: Procesadores similares a un Intel core pentium o superiores
3. Tarjeta gráfica: Similar o superior a la propia a una Intel integrada

4.3 Inventario de las herramientas de desarrollo

A continuación mencionaremos y añadiremos una pequeña descripción a las herramientas de desarrollo que hemos usado

1. Unity 2019.3.8f1 → Para el desarrollo del juego
2. OpenOffice Writer → Para la creación de documentos
3. OpenOffice Calc → Control de tareas realizadas
4. GitHub → Repositorio del proyecto con control de versiones
5. Logic Pro x → Música del videojuego
6. Paint 3D → Creación de imágenes de fondo y logo.
7. Moviemaker y Adobe Premier Pro X → Realización de los vídeos internos del juego como de presentación o trailer.

4.4 Inventario de los assets y recursos del juego

A continuación mencionaremos los distintos assets y recursos que se han usado para el desarrollo del juego.

1. Música menús y videoclip intro → Realizado en Logic Pro X por Laia Guilanyà Jané
2. Música en referencia a Anvein y usada para niveles exteriores → Realizada con Logic Pro X por Laia Guilanyà Jané

3. Música en referencia a los villanos y usada para niveles de castillo → Realizada con Logic Pro X por Laia Guilanyà Jané
4. Sprites Anvein → Obtenidos de la Unity Asset Store: https://assetstore.unity.com/packages/2d/characters/knight-sprite-sheet-free-93897
5. Sprites Decano → Obtenidos de la Unity Asset Store: https://assetstore.unity.com/packages/2d/characters/rpg-wizard-pack-143580#description
6. Sprites Tesla y Entrópico → Obtenidos de la Unity Asset Store: https://assetstore.unity.com/packages/2d/characters/dragon-warrior-free-93896
7. Sprites Bandidos → Obtenidos de la Unity Asset Store: https://assetstore.unity.com/packages/2d/characters/bandits-pixel-art-104130
8. Sprites Monstruos → Obtenidos de la Unity Asset Store: https://assetstore.unity.com/packages/2d/characters/gothicvania-enemies-pack-2-152990
9. Sprites para recursos → Obtenido de la Unity Asset Store: https://assetstore.unity.com/packages/2d/gui/icons/pixel-art-icon-pack-rpg-158343
10. Sprites Mundo Exterior → Obtenido de la Unity Asset Store: https://assetstore.unity.com/packages/2d/environments/nature-pixel-art-base-assets-free-151370
11. Sprites Mundo Castillo → Obtenido de la Unity Asset Store: https://assetstore.unity.com/packages/2d/environments/medieval-pixel-art-asset-free-130131
12. Sprites Tienda → Obtenido de la Unity Asset Store: https://assetstore.unity.com/packages/2d/characters/gothicvania-town-101407
13. Sprites Trampas → Obtenido de la Unity Asset Store: https://assetstore.unity.com/packages/2d/environments/2d-death-traps-ice-free-24714

14. Sprites para UI → Obtenido de la Unity Asset Store: https://assetstore.unity.com/packages/2d/gui/basic-ui-elements-concrete-48833
15. Fuente para la letra → Obtenido de la Unity Asset Store: https://assetstore.unity.com/packages/2d/fonts/free-pixel-font-thaleah-140059
16. Resto de imágenes simples → Obtenidas de Google
17. Imágenes compuestas de diferentes objetos → Creadas con Paint3D
18. Efectos de sonido → Obtenidos de https://freesound.org/
19. Efectos de partículas → Creados con Unity con Sprites para materiales obtenidos de Google

La creación musical la hicimos con Laia Guilanyà Jané enfocado en la historia que quería contar el videojuego, por ello, tenemos una canción genérica donde podemos ver un tono más neutral, después tenemos una música muy reconocible, estilo a Zelda: Breath of the Wild, para poder identificar al héroe y por último, una canción más oscura y tensa para mostrar donde están los malvados, al estilo de Súper Mario Bros y el castillo de Browser.

La creación de imágenes con Paint3D, se han realizado con diferentes tonalidades de color y añadiendo el personaje principal para dar un toque a las pantallas de los distintos menús.

Por último, los efectos de partículas, se han realizado con el componente de Unity con el mismo nombre. Ello ha conllevado horas de ajustes para poder tener unas partículas decentes para mostrar el efecto deseado. Retocando varios puntos de cada apartado del mismo componente. Esto se ha utilizado para la creación de poderes de todos los personajes y desaparición de los villanos con un efecto de humo.

4.5 Esquema de arquitectura del juego

A continuación haremos un esquema de la arquitectura del proyecto en dos niveles. Primero a nivel estructural y de carpetas y después a nivel de Scripts y sus relaciones.

En el primer punto, partimos de que hemos trabajado con Unity y su estructura previa de carpetas para el funcionamiento del motor y con Git para el control de versiones. Partiendo de la estructura de Unity, nuestro trabajo de arquitectura empieza en la carpeta llamada “Assets”, donde aquí, se han creado varias carpetas diferenciadas por temas o conceptos. A continuación haremos un listado de estas y después una explicación de su contenido.

Listado de Carpetas:

- Animations → En esta carpeta encontramos varias carpetas, con el nombre de enemigos, jugador o trampas, donde guardamos el componente animator y las distintas animaciones de cada conjunto.

- Font → En esta carpeta encontramos las fuentes de letra usadas en el juego.

- Material → En esta carpeta encontramos los materiales usados para los sistemas de particulas.

- Music → En esta carpeta encontramos dos carpetas, una para guardar las 3 canciones principales del juego y otra para guardar los efectos de sonido que tiene el juego.

- Paletas → En esta carpeta encontramos las dos paletas usadas para crear los niveles.

- Prefabs → En esta carpeta encontramos varias carpetas para diferenciar conjuntos de prefabs del juego. Estos prefabs son GameObjects que usamos de forma recursiva y de esta manera podemos crear uno y repetirlo por los niveles varias veces, o si estos están en distintos niveles. Tenemos los niveles de enemigos, con todos los enemigos y sus poderes, la de items, con todas las monedas, la de player, con el jugador y sus poderes, la de scenes, con prefabs genéricos, como el GameManager, la música background, la tienda, etc. Y por último tenemos la carpeta Traps, con todas las trampas.

- Resources → En esta carpeta encontramos los distintos json para hacer el control de guardado de partida, saber que niveles dejamos repetir al jugador o que poderes tenemos activos. Más adelante se hará una explicación de cada json.

- Scenes → En esta carpeta encontramos cinco carpetas divididas por tipo de pantalla, donde tenemos la carpeta de genericLevelScene con las pantallas de GameOver y FinishLevel. También tenemos una carpeta por cada bloque de niveles, es decir, por el bloque Intro, Tesla, Entrópico y Decano. Por último tenemos una carpeta con las pantallas de menú, como opciones, título, etc.

- Scripts → En esta carpeta encontramos todos los Scripts del juego. Cada Script tiene un nombre significativo, aunque más adelante se explicará un poco cada script y su función.

- Sprites → En esta carpeta, también encontraremos varias carpetas, divididas por tipo de Sprites, como por ejemplo los sprites de los personajes, cada uno en su carpeta o de los items,

o para la UI, etc. Básicamente se ha estructurado para poder encontrar cada sprite de un modo más rápido.

- Vídeos → En esta carpeta encontraremos el vídeo inicial del juego, que hace introducción al mundo y la historia del juego.

En este segundo punto de la arquitectura veremos un poco la jerarquía de los scripts y su uso. También explicaremos los json que usamos para guardar partida. Al tener 28 scripts, por algunos pasaremos un poco por encima, pero siguiendo la estructura explicativa de las carpetas, explicaremos los scripts en formato de listado, para tener un poco mejor de visión. Al principio explicaremos los dos scripts más globales y los json y después el GameManager. A partir de este punto, iremos explicando los demás sin un orden establecido.

Listado de Scripts por bloques:

Bloque Genérico:

- LevelVariables → Script que contiene todas las variables globales que usamos durante el juego. También tenemos variables para poder controlar cambios entre distintos scripts.

- managerJsonClasses → Fichero que contiene todas las clases para guardar en un json. Para poder tener el control de guardado de partida del jugador y información de cada nivel.
- Json → Tenemos quatro json que comentaremos a continuación:
  - powerUps.json → guardamos que power ups tiene activo el jugador
  - musicCongig.json → guardamos la configuración de la música y los efectos de sonido.

    - guardar.json → Guardamos el nivel del juego actual que se encuentra el jugador, el dinero acumulado y si tiene algún producto en el inventario (manzana, jamón o cerveza)
    - levelsInfo → Guardamos información de todos los niveles de la siguiente forma:
        - levelName: → nombre del nivel
        - indexLevel: → indice del nivel en Unity
        - showLevel: → Si mostramos el nivel en la pantalla de selección de nivel para volver a jugarlo. Solo se mostraran los niveles que ya hayamos conseguido superar.
        - GoldMoney, silverMoney, copperMoney → indicamos las monedas de cada tipo que tiene el nivel.

- GameManager → Controlador principal del juego dentro de los niveles. En este script controlamos el menú de Pausa y la pausa del juego, también hacemos guardado y control de variables globales. En los niveles de los Bosses finales, controlamos el inicio del juego para que el personaje final no se mueva.

- backgrounMusic → Controlador de la música de fondo de cada nivel.

Bloque Jugador(Anvein):

- playerManager → Este script controla todas las acciones del jugador principal Anvein. Aqui tenemos un código ordenado por nombres y funciones. En algunas funciones hay un poco de desorden con demasiados if, esto sería un punto a corregir para el futuro, también dividir el script en diferentes para no tener tantos if en el update, por ejemplo.

- cameraManager → Control de la camera que sigue al jugador

- playerBagManager → Control del inventario del jugador

Bloque Enemigos básicos y bandidos:

Estos Scripts se podrían haber unificado en un script global para las mismas funciones y scripts individuales con maquina de estados para cada uno de los estados en los que se encuentra el npc. Es un punto de mejora para el futuro.

- enemyBasicManager → Este script controla todos los enemigos básicos a excepción del amphibeo. Estos enemigos tienen una pequeña maquina de estados con if's, que hacen su movimiento o que reciben daño. Si tocan al jugador le hacen daño. Destacar, que el movimiento se ha echo a traves de raycast para comprobar cuando cambiar el sentido del mismo.

- amphibeanManager → Este script controla un único enemigo, que tiene un movimiento en vertical, como si fuera una serpiente. Es un script sencillo, donde aprovechamos los colliders de muerte del player para cuando caemos en los niveles para hacer el cambio de sentido.

- banditManager → Este script controla los bandidos. Estos enemigos tienen una IA un poco más profunda, y persiguen el jugador si lo detectan y atacan con la espada. También tiene una maquina de estados a base If's en el update.

Bloque Tesla, Entrópico y Decano:

Igual que en el bloque anterior, también se podrían haber modificado algunas funciones y tener una maquina de estados con ficheros en vez de, en este caso, un switch case.

- magicianManager → Controla los enemigos Tesla y Entrópico. Son scripts que persiguen al jugador hasta cierta distancia y atacan. También atacan cada cierto tiempo, random entre dos valores, donde lanzan bolas de su poder. Esto es para poder hacer un combate más realista entre jugador e IA. La IA la especificaremos más en el siguiente apartado.

- decanoManager → Controla al enemigo Decano. Tiene una IA y movimientos similares a los magos, con la diferencia, que ahora tenemos un ataque estático que se ha definido como un tercer estado, ya que este no termina hasta que le hacen daño y solo se activa cuando la salud esta por debajo de cierto rango.

- enemyPower → Controla las bolas de energía que lanzan los 3 bosses.

- decanoStaticPower → Controla el poder estático del Decano, que es una lluvia de truenos en toda la pantalla.

Bloque Niveles del Juego:

- shopManager → Control de la tienda y de la compra de productos.

- GameOver → Control de la pantalla del GameOver.

- finishLevel → Punto del nivel donde detectamos que es el final, añadimos animación del jugador corriendo y paramos movimiento de la cámara.

- finishLevelManager → Control de la pantalla de nivel superado.

- helpsManager → Control de los mensajes de ayuda del primer nivel. Donde damos nociones de los movimientos y ataques al jugador.

- startFinishLevel → Control del inicio y final de los niveles con los bosses. Tenemos un inicio de Anvein entrando corriendo y un texto del malo y al final abrimos un cofre con una llave y tenemos otro texto informándonos del poder conseguido.

- videoController → Control del vídeo de introducción a la historia antes del primer nivel

- arrowManager → Control de la trampa que lanza flechas. Desde el componente del script podemos modificar hacia que lado lanza la flecha, con que velocidad y cada cuantos segundos lanza una flecha. De esta manera, cada lanzador de flechas es distinto entre los niveles.

Bloque pantallas del menú:

- logoScript → Control de la pantalla del logo.

- titleScript → Control y animaciones de la pantalla del titulo.

- MainMenuScript → Control de la pantalla del menú principal y punto de guardado y capturar el estado inicial de cada inicio de juego.

- creditsManager → Control de la pantalla de créditos

- optionsManager → Control de la pantalla de opciones, con dos pestañas, una para la música y sonido y la otra para ver los

controles del juego. Controla también si la música esta disponible o no.

- selectLevelManager → Control de la pantalla de selección de nivel, para poder repetir niveles con más poderes o para poder conseguir todas las monedas de dicho nivel. Separada por bloques. Solo accesible al nivel una vez lo hemos superado siguiendo la historia.

4.6 Inteligencia Artificial del juego

En este punto explicaremos la Inteligencia Artificial que se ha instaurado en el juego. Hablaremos de forma global de esto, ya que hay partes comunas en ciertos enemigos y especificaremos aquellas que no lo sean.

El primer punto importante de la IA es el movimiento de los personajes, que tenemos dos tipos, movimiento con cambio de sentido antes de caer al vació y persecución del jugador. El primer caso lo controlamos con un raycast comprobando que no tocamos ningún colider y entonces cambiamos de sentido al jugador. Esto lo tienen los enemigos básicos y los bandidos. También perseguimos al jugador de dos maneras distintas, con los bandidos lo detectamos con un collider, y si el jugador entra lo perseguimos y si sale volvemos al movimiento normal. En cambio, los enemigos finales, nos persiguen todo el rato, a través de que conocen nuestra posición.

Ahora tenemos IA de control distinto, por ejemplo, para los bandidos lo controlamos a través de if's en el update, ejecutando un ataque, persiguiendo o moviéndonos normal. En este caso, el ataque se encuentra dentro de la acción de perseguir, donde, si la distancia es correcta atacará con la espada cada n segundo dentro de un rango. En cambio, los bosses tienen diferenciado el perseguir y el atacar con un switch case, una pequeña maquina de estados dentro de un mismo script. Este controla si esta atacando, o persiguiendo. El ataque de los

magos finales, es random entre distintos ataques y dependiendo de la distancia, si la distancia es corta lanzan un ataque físico, en cambio, si se encuentra lejos lanzan una bola de su poder. Por último, el decano, tiene 3 ataques mágicos y ninguno fisico, dos ataques que se van turnando lanzando bolas de energía o creando fuego a los pies del jugador y un ataque especial, que cuenta como un estado nuevo, donde crea una tormenta rayos.

Como podemos ver, tenemos una IA pequeña de maquina de estados echa en un mismo script. Este punto, tiene partes de mejora, tal y como hemos comentado, haciendo una maquina de estados con varios scripts y tener un mejor control de cada estado. También en un futuro, me gustaría introducir en juego de este estilo un IA de aprendizaje, donde el enemigo consiga una autonomía para elegir que acción hacer en cada caso.

5. Diseño de niveles

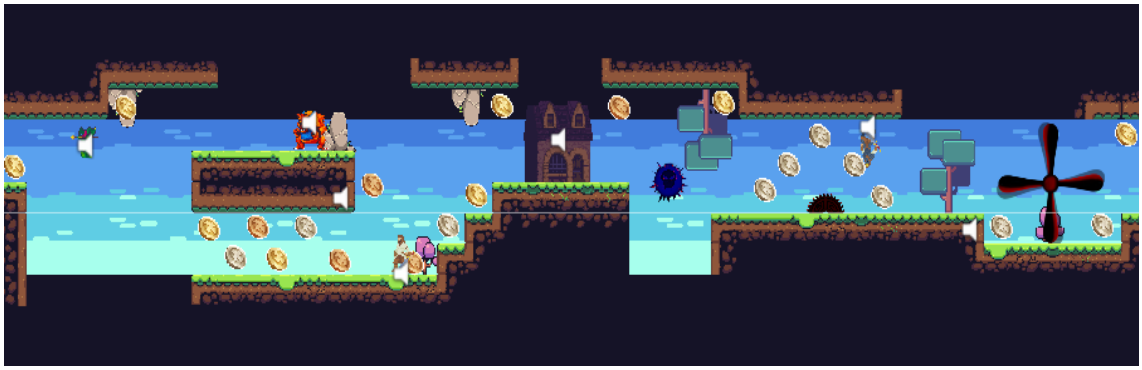
El juego actualmente tiene 10 niveles, un nivel inicial y 9 niveles, 3 por cada enemigo final. Dentro de los bloques de cada enemigo, tenemos un nivel exterior, un nivel castillo y el nivel de la sala del trono contra el mago final.

Primero os haré una pequeña explicación del nivel inicial y después un ejemplo de cada nivel de uno de los bloques, ya que la mayoría son similares.

El nivel introductorio, es le nivel más corto del juego similar a los niveles exteriores, donde vamos introduciendo diferentes aspectos del juego, como trampas, enemigos y la tienda. También nos encontramos pequeños postes, donde nos indican como hacer algunos movimientos o ataques básicos.

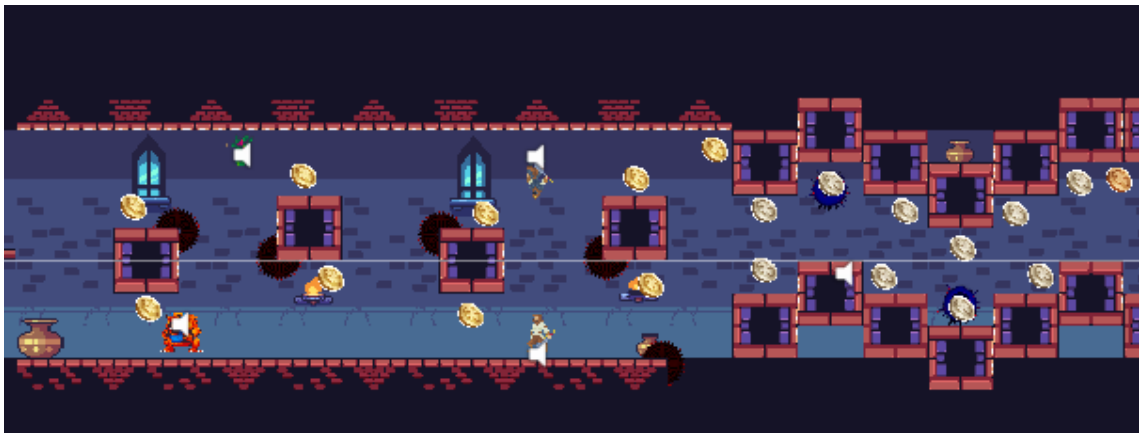
A continuación explicaremos un nivel de cada tipo y os mostraremos imágenes significativas del mismo.

Nivel Exterior:



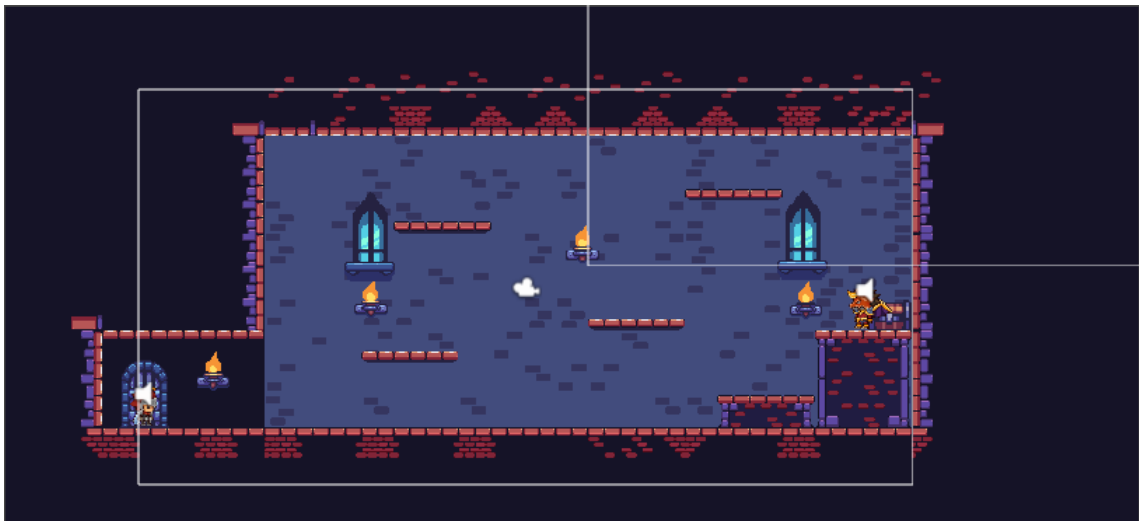
Como podemos ver, todos los niveles exteriores tienen la estructura de zonas de suelo y zonas de techo, para poder aprovechar la mecánica del cambio de gravedad, mecánica principal del juego. También jugamos con las trampas y enemigos para hacer pasar por varios sitios al jugador. Con monedas también conseguimos este afecto, ya que caminos sin salida, como el que vemos en la pantalla, solo vamos para recoger todas las monedas. Los enemigos también están tanto en el suelo como en el techo del mapa.
También vemos, que las tiendas están por la mitad del nivel, para ayudar al jugador a recolectar productos.

Nivel Castillo:



En el nivel de castillo vemos una estructura similar a los niveles exteriores, pero con más trampas y enemigos a lo largo del nivel. También vemos una estructura típica, que gracias a la mecánica del juego nos obliga a pasar por muchos sitios para poder avanzar.
Tanto los niveles exteriores como los de castillo, vemos como se propone una jugabilidad ágil y rápida para una mayor diversión y reto.

Nivel Sala del Trono:



En este nivel, vemos que es una única sala, con la cámara estática. En este nivel hay peldaños para poder escibar los ataques y huir del malvado mientras planeamos nuestro ataque. El villano nos seguirá por toda la pantalla, incluso cambiando su propio sentido de la gravedad, haciendo una bonita pelea final. También vemos un cofre, donde

empieza el villano, que contiene los manuscritos del poder que queremos conseguir.

Como podemos observar, todos lo niveles se han echo pensando en la mecánica principal del juego, para tener que aprovechar dicho poder y conseguir un modo distinto de jugar plataformas 2D.

6. Manual de usuario

6.1 Requisitos Hardware

Plataforma PC con sistema operativo Windows, Linux y Mac.

6.2 Ejecución del juego

Para ejecutar el juego, nos descomprimimos la carpeta en cualquier directorio y ejecutamos el archivo AnveinGameUnity.

7. Pruebas de usuario

Se han realizado pruebas de usuario con varios amigos con las s siguientes conclusiones:

1. Niveles muy divertidos
2. Necesidad de crear un sistema de mana, que controle los ataques mágicos
3. Necesidad de un inventario para guardar la comida comprada
4. Niveles complicados
5. Respuesta de los ataques enemigos
6. Tiempo entre ataques del jugador, para alinear con las animaciones

Con estás pruebas, he podido equilibrar un poco el juego y añadir dos funcionalidades nuevas, que no se habían contemplado al inicio de la creación del juego.

Con esto, podemos concluir, que aunque han sido pocas y no muy profesionales, las pruebas han servido para ver errores que el propio desarrollador no ve o no encuentra en sus propias pruebas. También atrae nuevas ideas para un acabado más completo.

8. Conclusiones

Para terminar el trabajo explicaremos un poco la experiencia obtenida y las lecciones aprendidas en el. También haremos un repaso de los objetivos conseguidos y de los que no, la metodología usada durante el desarrollo y por último la líneas de trabajo futuro.

Para empezar, en mi opinión, el trabajo me ayudado para ver la cantidad de trabajo y detalles que tiene un videojuego y que lo hace un buen producto. Pequeños detalles que no se ven a simple vista, pero una buena música de fondo, unos buenos dibujos y animaciones, una presentación vistosa, incluso un buen trailer, puede hacer que tu juego este más valorado que sino tuviera estás cosas. También se aprende a programar con más paciencia y de forma ordenada, al ser un proyecto grande, la necesidad de tenerlo todo ordenado y con nombres auto-explicativos, como un buen código comentado, que en este no es un buen ejemplo, pero necesario, sobretodo si se trabaja en grupo. Por último, hemos aprendido buenas formas de trabajo y los errores que comentaremos a continuación, nos dan experiencia para los siguientes.

Para empezar en los objetivos planteados, diremos que a nivel de jugabilidad, mecánicas y dinámicas se han cumplido todos los objetivos, al mismo nivel que de poder tener un producto finalizado, que en mi opinión, este lo es. Por otro lado, no he cumplido mis propios objetivos de un código perfectamente estructurado con programación orientada a objetos y en clases más globales, supongo que dada mi poca

experiencia en esta programación es un camino aprender, pero me hubiese gustado tener un código más estructurado, sobretodo en los scripts de los enemigos o del propio jugador. Tampoco he cumplido el objetivo de comentar el código, cosa que he aprendido a tener que hacerlo desde el primer segundo que empezamos a escribir código, al menos explicando que queremos hacer con esa función. Pero en lineas generales, estoy muy contento y satisfecho con el producto final.

La metodología de desarrollo ha sido la conocida como scrum, planteando sprint de dos semanas durante el semestre, para poder hacer tareas e ir avanzando el juego. Está metodología ha servido mucho para no tener que hacer el trabajo al final y poder disfrutar de su desarrollo. Incluso, nos ha servido para hacer pruebas con beta testers y poder añadir dos funcionalidades iniciales, que antes no se habían contemplado. De la misma manera, que poder ver que la cantidad de niveles planteadas al inicio, no eran viables para el desarrollo en este tiempo empleado. Tampoco hemos podido realizar todos los clips internos del juego que hubiésemos deseado, pero de todas formas, ha quedado un buen juego para el disfrute del público.

De cara el futuro, ya tenia cosas pensadas, como re-dibujar todos los personajes y el mundo, para poder usar unos assets propios y únicos. También añadir más niveles y clips para una mejor explicación de la historia. Incluso hacer un mini-libro de historia del juego, para poder hacer una saga en el futuro, tal y como se intuye en el nombre, Las Historias de Anvein. Por este motivo, estoy buscando un pequeño equipo para poder hacer un producto mejor y sentirlo todavía más nuestro. En esta linea, y dada mi vocación encontrada en este máster, me gustaría crear mi propio motor para desarrollar más juegos del mundo de Anvein, esté último punto es mi gran ilusión.

Para terminar, me gustaría comentar que gracias a este proyecto, he crecido como programador y me ha motivado a seguir luchando para llegar a crear mis propios juegos, espero llegar a tener mi pequeño estudio. También, al enseñar el trailer a diversas personas y recibir

buenas reseñas, me ha motivado y convencido de haber conseguido un buen juego, mi primer objetivo desde hace unos tres años.

9. Glosario

GDD → Game Design Document

IA → Inteligencia Artifical

Boss/Bosses → Enemigos finales del juego

Unity → Motor gráfico

PC → Ordenador

Assets → recursos usados para el videojuego

5. Bibliografía

No hay bibliografía en este trabajo.