## 1. Índice

1. Índice 2

2. Documento Descripción del proyecto. 3

2.1. Contexto del proyecto. 3

2.1.1. Ámbito y entorno. 4

Macro entorno: 4

Microentorno: 5

2.1.2. Análisis de la realidad. 6

2.1.3. Solución y justificación de la solución propuesta. 6

2.1.4. Destinatarios. 6

2.2. Objetivo del proyecto en castellano e inglés. 7

2.3. Marco legal. 10

3. Documento de Acuerdo del proyecto. 11

3.1. Requisitos funcionales y no funcionales. 11

3.2. Tareas. 11

3.3. Metodología a seguir para la realización del proyecto. 11

3.4. Planificación temporal de tareas. 11

3.5. Presupuesto (gastos, ingresos, beneficio). 11

3.6. Contrato/Pliego de condiciones. 11

3.7. Análisis de riesgos. 11

4. Documento de análisis y diseño. 11

4.1 Análisis y diseño de la arquitectura de la aplicación. 11

4.1.1 Tecnologías/Herramientas usadas y descripción de las mismas. 11

4.1.2 Arquitectura de componentes de la aplicación. 11

4.2 Modelado de datos. 11

4.3 Análisis y diseño del sistema funcional. 11

4.4 Análisis y diseño de la interfaz de usuario. 11

5. Documento de implementación e implantación del sistema. 11

5.1. Implementación. 11

5.2. Instalador. 11

5.3. Pruebas. 11

6. Documento de cierre. 11

6.1. Documento de instalación y configuración. 11

6.2. Manual de usuario. 11

6.3. Resultados obtenidos y conclusiones. 11

6.4. Diario de bitácora. 11

6.5. Temporalización y desviación sobre la planificación inicial. 11

7. Bibliografía. 11

8. Anexos. 11

## 2. Documento Descripción del proyecto.

##### 2.1. Contexto del proyecto.

El objetivo de desarrollar un videojuego como TFG es una idea que surgió tiempo después de la primera presentación de proyecto. En ambos casos, nos dimos cuenta de que el proyecto elegido no nos motivaba a dedicarle todo el tiempo libre que disponíamos, además de que era inviable aprender dos tecnologías nuevas en el tiempo que teníamos.

Ante esto, decidimos plantear un nuevo proyecto que nos motivase y al que probablemente pudiéramos darle mucha más calidad que al respectivo proyecto original.

En cuanto planteamos esto, vimos que teníamos una idea muy similar para desarrollar, y como hemos trabajado múltiples veces anteriormente en equipo, tenemos una mecánica de trabajo adaptada que nos resulta cómoda, por lo que era probable que entre los dos pudiéramos plantear y desarrollar un proyecto mucho más complejo y completo que haciéndolo por nuestra cuenta. Llegando a la conclusión de que era mucho más efectivo realizar un trabajo conjunto de una calidad mucho mayor a la que tendríamos si lo hiciéramos por separado.

El planteamiento es de un videojuego en 2D realizado en Unity, una plataforma que ya habíamos tocado con anterioridad. Además de tener una serie de ideas muy similares, lo cual ahorra problemas en el desarrollo.

Este proyecto se realiza bajo el marco de un trabajo de final de grado o TFG, el cual tiene un plazo aproximado de tres meses, entre abril y junio.

A fecha de lunes, 8 de abril, el proyecto nuevo está a la espera de aprobación del departamento.

##### 2.1.1. Ámbito y entorno.

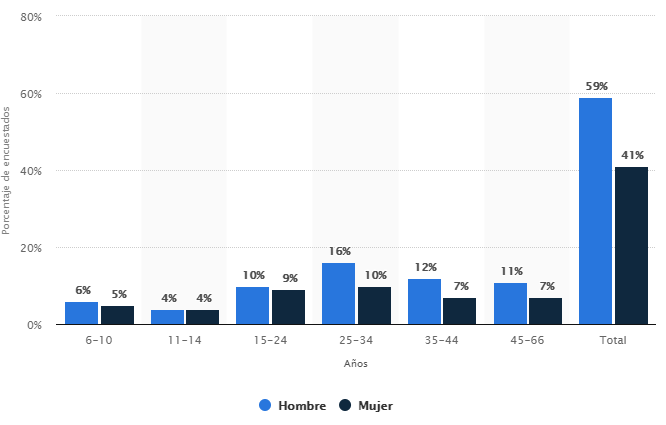
Análisis de Macro Entorno y Micro entorno:

##### Macro entorno:

La población adulta está comenzando a consumir cada día más productos de entretenimiento como los videojuegos, esto es consecuencia de que la población actual se ha criado junto con los juegos y la tecnología y, por tanto, son más abiertos al consumo y ampliar el conocimiento que tienen de esta. Como consecuencia, el público en un futuro próximo estará más interesado en adquirir productos relacionados con este sector.

Por otra parte, demográficamente la población mayor está aumentando poco a poco y disminuyendo la natalidad, lo cual puede afectar a todos los mercados a corto plazo, pero no a largo plazo porque la población restante estaría adaptada al consumo de esta tecnología.

La aparición de nuevas leyes referentes a la censura del contenido digital podría afectar al desarrollo del videojuego o al mantenimiento de este, estando como ejemplo, la aplicación del artículo 13 en la unión europea. Esta podría afectarnos en cuanto a copyright y tener que proteger la aplicación ante posibles robos de propiedad intelectual etc.

La política tanto española como europea, afectaría al videojuego en todos los aspectos económicos, ya que cualquier impuesto tecnología aumentaría los gastos y por tanto tendríamos que estimar un beneficio diferente por el desarrollo, además nos condicionaría los posibles riesgos a tomar.

Es importante tener en cuenta la integración del público femenino en el mercado de los videojuegos, siendo cada vez más significativo.

En la parte derecha podemos ver el porcentaje de jugadores según edad y género.

https://es.statista.com/estadisticas/481369/jugadores-de-videojuegos-en-espana-por-edad-y-genero/

Podemos ver que la presencia femenina es menor que la masculina en todas las edades, pero en comparación con hace unos años, la diferencia es casi nula, por lo que hay que prever que en los próximos años los niveles serán iguales o casi iguales.

##### Microentorno:

Los potenciales clientes serian principalmente cualquier persona con la capacidad de jugar un videojuego sin importar su edad o género.

La edad preferente por lo visto es 25 a 34 años para el consumo de videojuegos, con un 26% del volumen total, por lo que en los próximos años aumentara el grupo de 25 a 34 años con una nueva generación de jugadores.

Teniendo en cuenta la temática musical 8 bit y el estilo retro con pixelart, atraeríamos a un público algo mayor, entre los 25 y los 45 años, además, la velocidad de juego y la re-jugabilidad, haría de este un juego que podría moverse entre los videojuegos de Youtube o Twitch, por lo que ampliaríamos la edad de venta a un público que ronde entre los 13 y los 50 años.

Es importante participar en convenciones y grandes eventos, además de mantener un contacto continuo por medio de redes sociales con posibles contactos del sector. Estos hilos de amistad pueden derivar en escenarios de venta muy influyentes para el proyecto y llegar a realizar un volumen mucho mayor de ventas.

Como competidores entenderíamos a cualquier empresa indie que saque un juego de la misma temática, puesto que los clientes tendrán que decidir cuál juego comprarse normalmente. En este tema no hay mucho que tocar, puesto que la cantidad de empresas indie que salen a competir al mercado cada año con muy numerosas, y preocuparse por cada una de ellas sería imposible.

Lo único que podríamos hacer para combatir esto, es sin duda, realizar un juego mejor que la mayoría y que atraiga al público por su originalidad o simplemente porque sea divertido o adictivo de jugar.

Hay empresas del mundo de los videojuegos que no consideras competidores, puesto que se mueven con unas cifras que no puede ni imaginar una empresa pequeña, por lo que no los tomamos en cuenta. Lo único que podríamos tener en cuenta de ellas, sería el hecho de mantener un perfil bajo hasta ganar fama, puesto que si desarrollas un género nuevo que guste mucho, sacaran un juego con un presupuesto mayor imitando al tuyo, y nos aplastarían.

No disponemos de proveedores que condicionen la producción, por lo que no tenemos que esperar plazos de entrega.

Aunque no sean proveedores, hay que tener en cuenta que, si encargamos una música, por ejemplo, para el videojuego, recurriríamos a royalty free music, y tardaríamos en encontrar una adecuada en la extensa biblioteca de internet.

##### 2.1.2. Análisis de la realidad.

Actualmente contamos con pocos medios y un equipo de dos personas con conocimientos algo superiores a los de un usuario medio.

En lo referente a los videojuegos, tenemos mucha experiencia como consumidor, pero no como desarrolladores.

Por otra parte, tenemos amplios conocimientos en diseño por parte de Daniel Yaravide, y amplios conocimientos de scripting por parte de Sergio Suñén, así que tendríamos que combinar correctamente nuestras habilidades para acelerar el tiempo de desarrollo del proyecto.

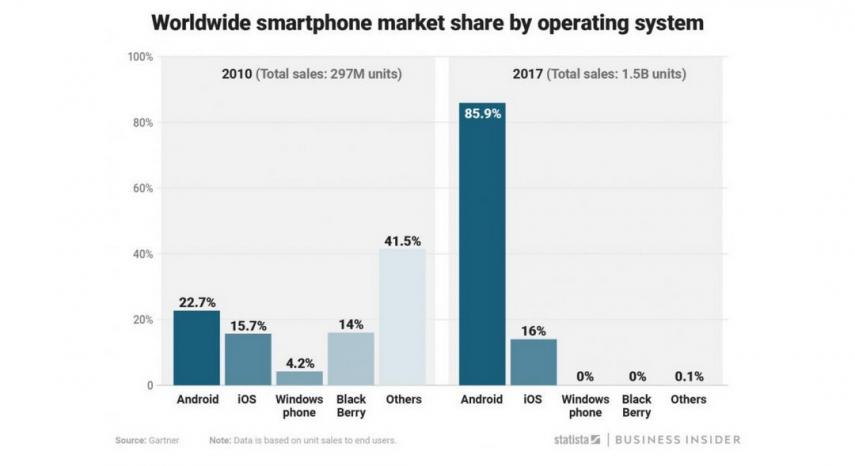
El plazo es relativamente escaso para los objetivos planteados, así que es posible que tengamos que recortar en cantidad de objetivos si vemos que no llegamos a tiempo.

Esto lo veremos cuando tengamos desarrollado una gran porción del proyecto o hayan pasado 2 meses desde el inicio del desarrollo, en este momento tendríamos que hacer balance del contenido desarrollado, el que falta por desarrollar y el tiempo que resta, valorando entonces si es viable llegar al final del desarrollo con todo lo propuesto, o recortar en ciertas partes.

##### 2.1.3. Solución y justificación de la solución propuesta.

##### 2.1.4. Destinatarios.

Como destinatario tendríamos a cualquier usuario de dispositivo móvil con la capacidad de jugar un videojuego, pero siendo más concreto, la edad preferente por lo visto es 25 a 34 años para el consumo de videojuegos, con un 26% del volumen total, por lo que en los próximos años aumentara el grupo de 25 a 34 años con una nueva generación de jugadores.

Teniendo en cuenta la temática musical 8 bit y el estilo retro con pixel art, atraeríamos a un público algo mayor, entre los 25 y los 45 años, además, la velocidad de juego y la re-jugabilidad, haría de este un juego que podría moverse entre los videojuegos de YouTube o Twitch, por lo que ampliaríamos la edad de venta a un público que ronde entre los 13 y los 50 años.

En la gráfica expuesta a la derecha, podemos observar el estado del mercado actual en cuanto a venta de Smartphones, por lo que podemos observar, Android domina las ventas en 2017 por un 85.9% de las ventas mundiales.

Esto nos beneficia puesto que nuestro principal cliente será el usuario de teléfonos Android.

##### 2.2. Objetivo del proyecto en castellano e inglés.

Realizar un videojuego de plataformas en 2 dimensiones, para su despliegue en plataformas móviles.

Como principal objetivo, tenemos que desarrollar un juego que sea fluido y re-jugable múltiples veces, por tanto, aunque el juego no tenga historia, el jugador tiene que notar que avanza, ya sea por la dificultad como por una puntuación adecuada.

El estilo de juego a desarrollar entra en la categoría de los denominados como ‘Rogue Like’, por definición, Roguelike es un subgénero de videojuegos de rol que se caracteriza por un rastreo de mazmorras a través de niveles generados de manera aleatoria, usualmente se desarrolla por turnos, gráficos basados ​​en tablas y la muerte reiterada del personaje del jugador.

Es imporante plantear un nuevo juego de esta categoría que se adapte a un sistema actual que ya funciona bien, pero que tenga su propio estilo y características que lo diferencien del resto.

Dividiremos los objetivos en diferentes partes:

**Visual / Sonido:**

Desarrollar o adquirir componentes visuales suficientes como para cubrir todas las animaciones referentes a un videojuego.

Sonido, como tenemos poca formación para crear música y sonidos, tendremos que adquirir música para:

* Niveles
* Bosses
* Pantalla de Inicio
* Muerte

Y sonidos varios para:

* Impacto
* Salto
* Daño
* Muerte
* Golpe de arma
* Disparo de arma
* Ataque enemigo
* Botones
* Carga
* Dash o Rueda (movimiento especial)
* Fuego
* Explosiones
* Crujido de madera
* Pasos en piedra
* Pasos en madera
* Sonidos varios de enemigos especiales
* Sonido ambiental

**Base de datos:**

Debemos desarrollar un sistema de base de datos orientado a un gran número de usuarios y que se actualice en tiempo real.

Esta debe ser ligera ya que enviara información a los jugadores y la recibe continuamente, y en caso de los dispositivos portátiles, significa un gasto de megabytes alto.

La base de datos no guardara ninguna información sensible del jugador.

Para esta base de datos utilizaremos FireBase, la cual tiene componentes coordinados con Unity y una serie de potentes SDK adaptables.

Además de guardar la información que necesitamos, FireBase se encargara de analizar datos de los jugadores a medida que juegan, como cuánto tiempo tardan en cada nivel o el tiempo que invierten en la app.



**Programación:**

Debemos programar scripts para generación semi-aleatoria de salas, es decir, los mapas, aunque estén predefinidos, tendrán salas con una variedad de aproximadamente 10 variaciones por tipo de sala, lo que dará lugar a:

5\*10 salas fáciles = 50 salas fáciles.

3\*10 salas difíciles = 30 salas difíciles:

2 salas de boss.

50 + 30 = 80 posibles combinaciones por mapa.

Con 8 mapas creados da lugar a 80 elevado a 8 posibles combinaciones de nivel, lo que equivale a aproximadamente 167772160000000 posibilidades de juego, no son infinitas, pero es casi imposible que te toque dos veces el mismo mapa.

Además de esta generación aleatoria, debemos programar la IA de los enemigos, que hará que los enemigos persigan al jugador sin quedarse atascados, o le golpeen si tienen la oportunidad.

Programar internamente la reacción entre los botones y la jugabilidad del juego, que respondan bien y rápido a las órdenes del jugador.

Otro reto a desarrollar es la IA de los Bosses o enemigos finales, ya que actuarán de forma diferente a los demás enemigos, o tendrán capacidades diferentes en cuanto a vida. Esto implica que tenemos que desarrollar un sistema de daños y una base predefinida de enemigos y como estos interaccionan.

**Reparto de tareas:**

Una distribución de objetivos correctamente planteada sería algo así:

***Visual + Sonido:***

***Daniel Yaravide****, se encargará de la recopilación y desarrollo del arte visual del juego además de la parte que se comunica con el usuario y con la que interactúa, conocido como HUD o User Interface. Además de la recopilación, se encargará de implementarlo correctamente dentro del videojuego.*

***Sergio Suñén****, se encargará recopilar, elaborar é de implementar los sonidos en los scripts de evento creados, para dar la sensación de comunicación con el usuario con cada gesto y movimiento, esto dará al usuario una inmersión mayor dentro del juego.*

***Base de datos:***

***Daniel Yaravide****, planteara la estructura de la base de datos y la implementara en la página correspondiente de FireBase. Además, añadirá en los scripts correspondientes la comunicación entre Unity y FireBase.*

***Sergio Suñén****, implementara los recopiladores de información en los scripts para enviarlos a la base de datos en el momento necesario. También añadirá los scripts de comunicación con FireBase correspondientes.*

***Programación:***

***Daniel Yaravide****, desarrollara los scripts correspondientes al movimiento del jugador y la interacción entre el usuario y el HUD. Esta parte la desarrollará utilizando Unity, Visual Studio y Photoshop.*

***Sergio Suñén****, desarrollara los scripts correspondientes al movimiento de los enemigos, su inteligencia artificial y su path finding. Esta parte la desarrollará con Unity y Visual Studio.*

***En conjunto*** *desarrollarán la generación aleatoria y las opciones referentes al juego como pueden ser la dificultad o la modalidad.*

Este reparto de tareas es básico, pero utilizaremos un sistema basado en listas de tareas implementado por la plataforma Trello. Esta se incluirá en el presente documento tan pronto como esté terminada.

##### 2.3. Marco legal.

Para cumplir con el marco legal correspondiente, tendríamos que aceptar los términos y condiciones de la plataforma distribuidora que en este caso es Google Play.

Estos son los términos legales a cumplir por parte del desarrollo para pasar el filtro de Google Play:

**Acceso al Contenido y su uso**. Podrá usar Google Play a fin de explorar, encontrar, mirar, transmitir o descargar Contenido para su dispositivo móvil, computadora, TV, reloj o cualquier otro dispositivo compatible ("**Dispositivo**"). Para usar Google Play, necesitará un Dispositivo que cumpla con los requisitos de compatibilidad y de sistema del Contenido correspondiente, que tenga una conexión de Internet que funcione y cuente con el software compatible. La disponibilidad del Contenido y las funciones variarán según el país y es posible que no todo el Contenido o las funciones estén disponibles en su país. Asimismo, parte del Contenido podrá estar disponible para compartir con miembros de su grupo familiar. Es posible que Google ofrezca Contenido o que cierto Contenido esté disponible mediante terceros que no estén afiliados con Google. En ese caso, Google no se responsabilizará ni responderá por ningún Contenido disponible en Google Play que no proceda de Google.

**Restricciones de edad.** Para usar Google Play, deberá tener una cuenta de Google válida ("**Cuenta de Google**"), que estará sujeta a las siguientes [restricciones de edad](https://support.google.com/accounts/answer/1350409?hl=es-419). Si se lo considera menor de edad en su país, deberá contar con el permiso de su padre o tutor legal para usar Google Play y aceptar las Condiciones. Deberá cumplir con cualquier restricción de edad adicional que pueda aplicarse al uso de funciones o Contenido específicos de Google Play. Los administradores y miembros del grupo familiar también deberán cumplir con requisitos adicionales.

**Tarifas de terceros.** Usted será responsable de cualquier tarifa de acceso o datos que se incurra mediante terceros (como su proveedor de Internet o proveedor de telefonía celular) en relación con su uso y visualización del Contenido y Google Play.

**Actualizaciones.** Es posible que Google Play, las bibliotecas de asistencia relacionadas o el Contenido requieran actualizaciones, como correcciones de errores, mejoras de funciones, complementos faltantes y versiones nuevas (en conjunto, "**Actualizaciones**"). Dichas actualizaciones podrán ser necesarias para que pueda usar Google Play o bien acceder, descargar o usar el Contenido. Al aceptar estas Condiciones y usar Google Play, acepta recibir dichas Actualizaciones automáticamente. Asimismo, también podrá administrar las Actualizaciones de cierto Contenido mediante Configuración en Google Play. Sin embargo, si se determina que la Actualización solucionará una vulnerabilidad de seguridad crítica relacionada con el Contenido, dicha Actualización podrá completarse independientemente de la configuración de la Actualización en Google Play o su Dispositivo. Si otra tienda de aplicaciones intenta actualizar el Contenido que se descargó inicialmente de Google Play, es posible que reciba una advertencia o que dichas actualizaciones se eviten por completo.

**Información sobre usted.** En las políticas de privacidad de Google, se explica cómo administramos sus datos personales y cómo protegemos su privacidad cuando usa Google Play. Es posible que Google deba compartir su información personal (como su nombre y dirección de correo electrónico) con los Proveedores a fin de ofrecerle Contenido o procesar sus transacciones. Los Proveedores aceptan usar esta información según sus políticas de privacidad.

Si es parte de un grupo familiar en Google Play, los miembros de su grupo podrán ver información determinada sobre usted. Si usted es el administrador de un grupo familiar en Google Play, los miembros que invite a unirse al grupo verán su nombre, foto y dirección de correo electrónico. Si se une a un grupo familiar como miembro, los otros miembros podrán ver su nombre, foto y dirección de correo electrónico. El administrador de su grupo familiar también podrá ver su edad y un registro de todas las compras que realice con la forma de pago familiar designada, lo que también incluirá una descripción del Contenido comprado. Si el Contenido está disponible para uso compartido con el grupo familiar y usted lo comparte con su grupo, todos los miembros podrán acceder al Contenido y ver que usted lo compró.

**Acceso no autorizado a las cuentas.** Deberá proteger la información de su cuenta y no compartirla con ninguna otra persona. No recopile datos personales de ningún usuario de Google Play ni de otros Servicios de Google mediante Google Play, incluidos los nombres de cuentas.

**Protección contra software malicioso.** Para protegerlo de software malicioso de terceros, URL y otros problemas de seguridad, es posible que Google reciba información sobre las conexiones de red de su Dispositivo, las URL potencialmente dañinas, el sistema operativo y las aplicaciones instaladas en su Dispositivo a través de Google Play o de otras fuentes. Google podrá advertirle si considera que una aplicación o URL no es segura, o podrá borrar o bloquear la instalación en su Dispositivo si toma conocimiento de que es dañina para los dispositivos, datos o usuarios. Usted podrá optar por inhabilitar algunas de estas protecciones en la configuración de su Dispositivo. Sin embargo, Google podrá continuar recibiendo información sobre las aplicaciones instaladas a través de Google Play, y es posible que se continúen analizando las aplicaciones instaladas en su Dispositivo mediante otras fuentes a fin de buscar problemas de seguridad sin que se envíe información a Google.

## 3. Documento de Acuerdo del proyecto.

##### 3.1. Requisitos funcionales y no funcionales.

Crearemos un juego del estilo Rogue Like, Mazmorras y plataformas 2D, altamente inspirado en juegos como The Binding of Isaac o Rogue Legacy.

El juego tendrá que disponer de aleatoriedad y re-jugabilidad, de modo que puedas repetir los niveles una y otra vez sin que te resulte repetitivo.

Al querer dar una historia al juego, el orden en el que se suceden las salas debe ser el mismo, así que, como solución a esto, habrá que dar aleatoriedad al interior de las salas, por lo tanto, cada sala cargará antes de entrar a esta lo que habrá dentro, siendo aleatorio todo lo que haya en el interior, desde plataformas a enemigos.

1. Deberá haber dos modos de juego, uno con generación completamente aleatoria, es decir, no tiene fin, cada X número de salas habrá una sala de boss, y otro modo con historia, en el que cada sala tendrá un orden especifico.
2. Debe haber un total de 80 habitaciones con sus entradas y salidas.
3. Debe haber suficientes variaciones del interior de las salas como para que no resulte repetitivo volver a jugar el juego.
4. Crear 4 personajes los cuales iras desbloqueando a medida que juegues.
5. Crear interacción entre el jugador y el mapa.
6. Que el jugador note su evolución mientras juega.

##### 3.2. Tareas.

Tareas a realizar a lo largo del trabajo, estas son unas tareas objetivo, pero pueden no ser añadidas todas al resultado final:

Visual:

1. Búsqueda de sprites para Pyro, Charger, Taco, Mirror.
2. Búsqueda de entorno para zonas fáciles.
3. Búsqueda de entorno para zonas difíciles.
4. Búsqueda de entorno para zonas de Boss.
5. Búsqueda de sprites para enemigos terrestres.
6. Búsqueda de sprites para enemigos voladores.
7. Búsqueda de sprites para enemigos finales o bosses.
8. Búsqueda de sprites para objetos.
9. Búsqueda de sprites para efectos.
10. Búsqueda de sprites para plataformas.
11. Búsqueda de sprites para iluminación de entorno.
12. Búsqueda de sprites para ataques de boses.
13. Búsqueda de sprites para ataques de enemigos.
14. Búsqueda de imágenes para botones.
15. Búsqueda de imágenes para interfaz.
16. Búsqueda de imágenes para ajustes.

Sonido:

1. Búsqueda de sonidos para ataque básico.
2. Búsqueda de sonidos para segundo ataque básico.
3. Búsqueda de sonidos para ataque fuerte.
4. Búsqueda de sonidos para ataque especial.
5. Búsqueda de sonidos para salto.
6. Búsqueda de sonidos para caída.
7. Búsqueda de sonidos para disparo.
8. Búsqueda de sonidos para daño a jugador.
9. Búsqueda de sonidos para daño a enemigo.
10. Búsqueda de sonidos para muerte de jugador.
11. Búsqueda de sonidos para muerte de enemigo (por enemigo).
12. Búsqueda de sonidos para ataque de enemigo.
13. Búsqueda de sonidos para dash.
14. Búsqueda de sonidos ambientales (Fuego, gotas, charcos…).
15. Búsqueda de sonidos para caminar.
16. Búsqueda de sonidos para recoger objeto.
17. Búsqueda de sonidos al pulsar botón.

Música:

1. Música por nivel (Hielo, Tierra, Agua, Fuego…).
2. Música para boss (una por boss).
3. Música para menú de inicio.
4. Música para muerte.

Scripting:

1. Script de movimiento de personaje.
2. Script de ataque básico de personaje.
3. Script de ataque mejorado de Pyro.
4. Script de ataque mejorado de Taco.
5. Script de ataque mejorado de Charger.
6. Script de ataque mejorado de Mirror.
7. Script de dash para Pyro.
8. Script de dash para Taco.
9. Script de dash para Charger.
10. Script de dash para Mirror.
11. Script de estadísticas del personaje.
12. Objeto de enemigo volador
13. Script de movimiento de enemigo volador.
14. Script de ataque de enemigo volador
15. Objeto de enemigo terrestre
16. Script de movimiento de enemigo terrestre.
17. Script de ataque de enemigo terrestre.

##### 

##### 3.3. Metodología a seguir para la realización del proyecto.

Es importante recalcar que somos dos personas desarrollando a la vez, así que un sistema de organización adecuado tiene que ser implementado desde el primer momento, como sistema de organización dentro del proyecto, hemos decidido adoptar un sistema basado en la metodología Kanban y fusionado con parte de la metodología Scrum.

Dispondremos tableros con tareas a realizar utilizando la herramienta de Trello, de uso compartido, online y principalmente de uso gratuito para este tipo de entorno.

En cuanto a la metodología Scrum, plantearemos una serie de Stand Ups diarios o cada dos días, en los que pondremos en común lo trabajado el día anterior, y lo que trabajaremos a lo largo del día en curso.

Añadiremos ejemplos de esta organización cuando acabemos de implementar esta tecnología, esto está planteado para estar acabado antes del miércoles 10 de abril.

##### 3.4. Planificación temporal de tareas.

|  |  |  |  |
| --- | --- | --- | --- |
| TAREA | dESCRIPCIÓN | TIEMPO ESTIMADO | REALIZADO POR: |
| planteamiento | Elegir el estilo de juego sobre el que trabajaríamos, si se desarrollaría sobre una historia lineal o sería re-jugable múltiples veces. Protagonistas, estilo del juego, música, diseño, velocidad de game play… | ±15 horas | Sergio Suñén & Daniel Yaravide |
|  |  |  |  |
| aprendizaje  (unity tutorials) | Es destacable en el desarrollo del proyecto, el tiempo dedicado a aumentar los conocimientos de la plataforma sobre la que trabaja el equipo. | ±60 horas | Sergio Suñén & Daniel Yaravide |
|  |  |  |  |
| Diseño de nivel | Diseñar un patrón de nivel que se repetirá continuamente, sin embargo, el contenido de las salas tendrá aleatoriedad. | ±1 hora | Daniel Yaravide |
|  |  |  |  |
| Requisitos de personajes y enemigos | Diseñar un sistema de combate tanto para el jugador como para los enemigos | ±2 horas | Sergio Suñén |
|  |  |  |  |
| DISEÑO de Personaje: pyro | Realización del aspecto visual y las animaciones de uno de los personajes para poder trabajar sobre él. | ±5 horas | Daniel Yaravide |
|  |  |  |  |
| script y animación de pyro | Realización del script de movimiento y el flujo de animaciones para el personaje principal, sirviendo como base para los demás personajes | ± 15 horas | Sergio Suñén |
|  |  |  |  |
| diseño entorno:  Mapa 4 | Diseño del mapa número 4, esta numeración se lleva a cabo por la posición de la puerta en la sala principal.  Creación de variantes de cada sala. | ±10 - 15 horas | Daniel Yaravide |
|  |  |  |  |
| Script de entorno y generación aleatoria | Scripting para la variación de las salas de forma aleatoria entre un grupo planteado previamente. Además se programaran los elementos destruibles del entorno. | ±3 horas | Daniel Yaravide |
|  |  |  |  |
| script y animación:  mimic | Scripting para generar una inteligencia artificial para un enemigo del tipo: cofre trampa. Este esperara a tener el jugador cerca para atacar. | ±5 horas | Sergio Suñén |
|  |  |  |  |
| Creación de los mapas 1,2,3,5,6,7,8 utilizando tilemaps | Crear mapas con formas determinadas en los que posteriormente se puedan añadir zonas aleatorias prediseñadas. | ±2 horas por mapa | Daniel Yaravide |
|  |  |  |  |
| rediseño del mapa 4 por razones tecnicas utilizando tilemaps | Rediseñar el mapa 4 para que tenga coherencia visual con los demás mapas, en esta caso utilizaremos tilemaps. | ±3 horas | Daniel Yaravide |
|  |  |  |  |
| Estructura de generación de escenas multiples | Generar elementos que carguen escenas dentro de la escena principal para tener únicamente cargado aquello que es necesario para el jugador en cada momento. | ±4 horas | Sergio Suñen |
|  |  |  |  |
| ia de los enemigos menores | Dar inteligencia y movilidad a los enemigos menores del juego, dándoles personalidad propia y animaciones. | ±30 horas | Sergio Suñen & Daniel Yaravide |
|  |  |  |  |
| ia de los enemigos principales /bosses | Dar patrones de movimiento a los enemigos principales del juego, aquellos a los que el jugador se enfrentara una vez por nivel y por tanto tienen que actuar diferente del resto | ±30 horas | Sergio Suñén & Daniel Yaravide |
|  |  |  |  |
| implementación de los enemigos en las escenas | Añadiremos los enemigos menores en las escenas que se generan dentro de las salas de modo que los enemigos se instancien al crearse la sala y no continuamente. | ±4 horas | Sergio Suñén & Daniel Yaravide |
|  |  |  |  |
| implementación de los bosses por nivel | Estos enemigos finales aparecerán únicamente en sus salas propias, teniendo animaciones e interacciones especiales, por lo que hay que tener en cuenta muchos factores de entorno. | ±8 horas | Sergio Suñén & Daniel Yaravide |

##### 3.5. Presupuesto (gastos, ingresos, beneficio).

GASTOS:

UNITY – 0 €

Unity resulta gratuito siempre que los beneficios sean inferiores a 100.000 € anuales, por lo que no nos costará ningún dinero mientras nuestro proyecto no supere esa cantidad de beneficios.

VISUAL STUDIO – 0 €

Dentro del marco de uso de Unity, Visual Studio resulta gratuito bajo las mismas condiciones.

SPRITES – 19.95 €

SPRITES & TILES para enemigos y mapa por 19.95 €

INGRESOS:

El juego se lanzará con un precio de 1,3 € así que esperamos un beneficio total de 3500 €.

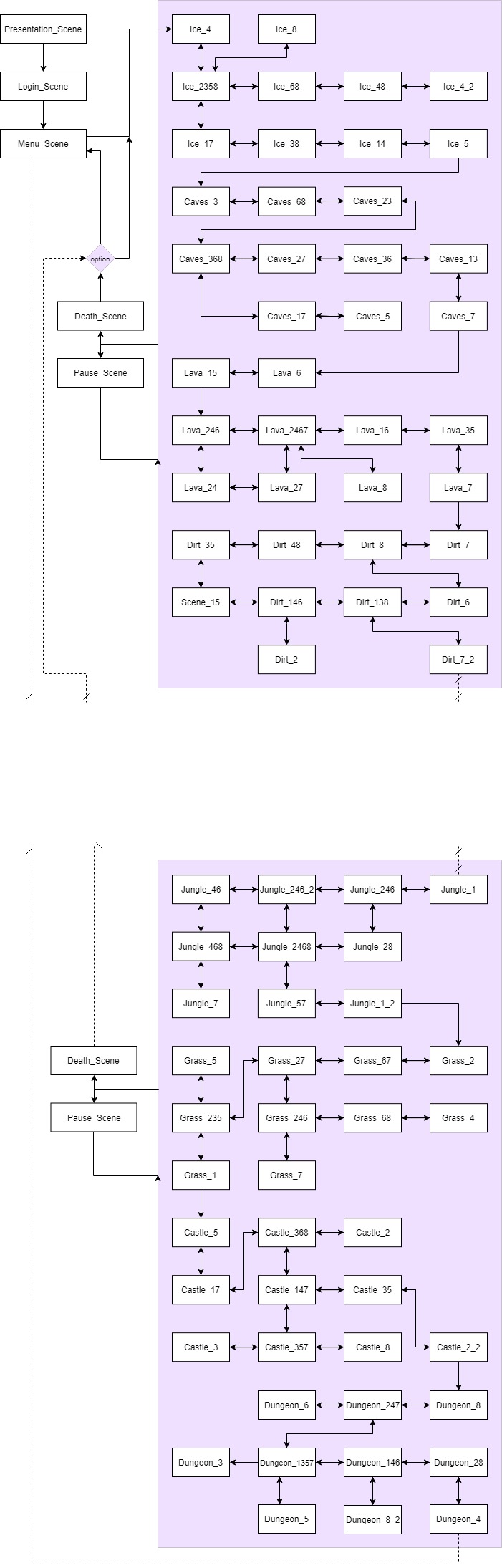
##### 3.6. Contrato/Pliego de condiciones.

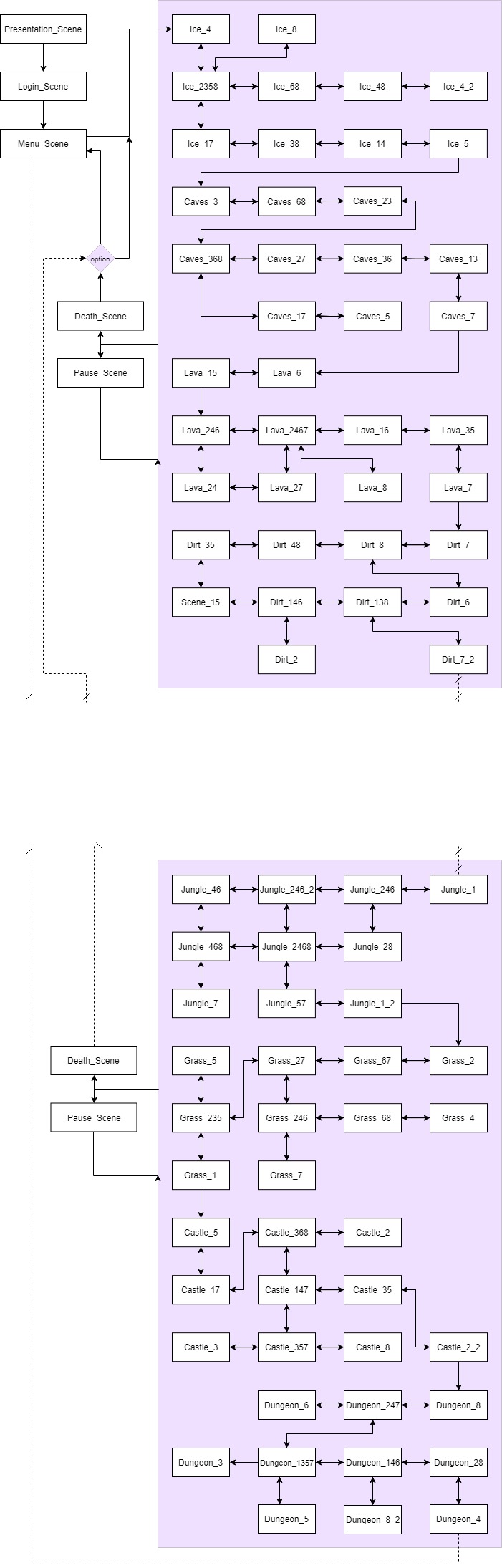
##### 3.7. Análisis de riesgos.

## 

## 4. Documento de análisis y diseño.

##### 4.1 Análisis y diseño de la arquitectura de la aplicación.





##### 4.1.1 Tecnologías/Herramientas usadas y descripción de las mismas.

Draw.io – Herramienta gratuita construida en base a Google Drive™ que permite crear tablas, diagramas de flujo, UML… Esta herramienta la hemos utilizado para crear los flujos de escenas, animaciones, clases…

C# - Hemos utilizado este lenguaje para programar sobre Unity, los scripts de este pueden ser desarrollados tanto en C# como en JavaScript, pero hemos decidido usar el primero.

Itch.io – Herramienta online para encontrar, gratis o pagando, un gran número de sprites y componentes para videojuegos tanto en 2D como en 3D.

Visual Studio – Herramienta de Microsoft para crear, editar y depurar código, esta herramienta esta predefinida por Unity para crear sus scripts.

Unity – Herramienta indispensable para crear nuestro videojuego actualmente, esta se adapta a la capacidad económica de los creadores, y sobre todo para juegos de bajo presupuesto como el nuestro.

##### 4.1.2 Arquitectura de componentes de la aplicación.

##### 4.2 Modelado de datos.

##### 4.3 Análisis y diseño del sistema funcional.

##### 4.4 Análisis y diseño de la interfaz de usuario.

## 5. Documento de implementación e implantación del sistema.

##### 5.1. Implementación.

##### 5.2. Instalador.

##### 5.3. Pruebas.

## 6. Documento de cierre.

##### 6.1. Documento de instalación y configuración.

##### 6.2. Manual de usuario.

##### 6.3. Resultados obtenidos y conclusiones.

##### 6.4. Diario de bitácora.

##### 6.5. Temporalización y desviación sobre la planificación inicial.

## 7. Bibliografía.

## 8. Anexos.