



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA (SEGOVIA)

Grado en Ingeniería Informática de Servicios y Aplicaciones

Desarrollo de un videojuego en un entorno
móvil con Unity3D
[The Dark World-The Age Of Darkness]

Alumno: Javier Garzón Merino

Tutor: Anibal Bregon Bregon

Miguel Ángel Martínez Prieto



DESARROLLO DE UN VIDEOJUEGO EN 2 D PARA ANDROID CON UNITY 3 D

Trabajo de Fin de Grado
Ingeniería Informática de Servicios y Aplicaciones
Julio 2017

Autor: Javier Garzón Merino
Tutor/es: Aníbal Bregón Bregón
Miguel Ángel Martínez Prieto

Índice General

Introducción	11
1.1 Motivación.....	12
1.2 Objetivos.....	14
1.3 Alcance	15
1.4 Herramientas Utilizadas	15
1.4.1 ¿Por qué Unity3D?.....	16
1.4.1 ¿Por qué Android?.....	17
1.5 Organización del documento	19
Estado del arte	21
2.1 Juegos de referencia.....	24
Introducción a Unity3D	31
Planificación y presupuesto	39
4.1 Metodología.....	39
4.2 Planificación	40
4.3 Presupuesto inicial.....	42
4.3.1 Presupuesto hardware	43
4.3.2 Presupuesto software.....	45
4.3.3 Presupuesto de desarrollo.....	45
4.3.3 Presupuesto total	46
4.4 Estimación de costes mediante COCOMO	47
4.4.1 Estimación costes hardware.....	50
4.4.2 Estimación costes software	52
4.4.3 Presupuesto de desarrollo.....	53
4.4.4 Presupuesto Total.....	54
4.5 Conclusiones.....	54
Análisis.....	55
5.1 Identificación de actores	56
5.2 Objetivos.....	56
5.3 Especificación de Requisitos	60
5.3.1 Requisitos Funcionales	60

5.3.2 Requisitos no Funcionales	85
5.4 Casos de uso	88
5.4.1 Requisitos Diagrama de Casos de Uso.....	88
5.4.2 Especificación de Casos de Uso	89
5.5 Árbol de características	98
5.6 Diagramas de secuencia.....	100
Diseño.....	104
6.1 Arquitectura lógica	104
6.2 Arquitectura física	105
6.3 Diagramas de clases de diseño	106
6.4 Diagramas de estados	109
Actualmente, la música del juego puede pasar por dos estados, encendido y apagado.....	113
6.5 Historia	114
6.6 Estética del juego.....	116
Implementación	137
7.1 Organización.....	137
7.2 Pantallas de juego	140
7.4 Personaje.....	151
7.5 Enemigos	156
Pruebas	159
Manual de Instalación.....	163
Manual de Usuario	171
Conclusiones.....	178
11.1 Futuras mejoras	179
Bibliografía & Webgrafía.....	181
Glosario	184
Contenido del CD-ROM	187

ÍNDICE DE TABLAS

Tabla 1. Presupuesto Inicial – Presupuesto hardware	44
Tabla 2. Presupuesto inicial – Presupuesto software	45
Tabla 3. Presupuesto inicial- Presupuesto de desarrollo	46
Tabla 4. Presupuesto inicial – Presupuesto Total	46
Tabla 5. Factores de coste COCOMO	48
Tabla 6. Factores de complejidad COCOMO.....	49
Tabla 7. Esfuerzo categorías de trabajo	50
Tabla 8. Estimación de costes hardware –COCOMO-.....	52
Tabla 9. Estimación de costes software –COCOMO-.....	53
Tabla 10. Presupuesto inicial – Presupuesto de desarrollo.....	53
Tabla 11. Presupuesto inicial – Presupuesto total	54
Tabla 12. Comparación resultados	54
Tabla 13. ACT-01 Jugador.....	56
Tabla 14. .Plantilla Ejemplo de tabla Objetivos.....	56
Tabla 15. OBJ-01 Diseño del videojuego	56
Tabla 16. OBJ-02 Creación de la GUI.....	57
Tabla 17. OBJ-03 Creación del sonido del juego	57
Tabla 18. OBJ-04 Implementación Multi-Juego.....	58
Tabla 19. OBJ-05 Creación Nivel “Infinito”	58
Tabla 20. OBJ-06 Creación de distintos niveles	58
Tabla 21. OBJ-07 Creación Modo Historia	59
Tabla 22. OBJ-08 Cargar progreso.....	59
Tabla 23. OBJ-09 Guardar Progreso.....	59
Tabla 24. OBJ-10 Creación del sonido de juego	59
Tabla 25. OBJ-11 Creación modo historia.....	60
Tabla 26. Ejemplo de tabla de Requisitos	60
Tabla 27. RQF-01 Seguir al jugador con la Cámara.....	61
Tabla 28. RQF-02 Colisionar con Objetos.....	61
Tabla 29. RQF-03 Gestor Animaciones	61
Tabla 30. RQF-04 Muerte Personaje	62
Tabla 31. RQF-05 Recoger Item/Cristal.....	62
Tabla 32. RQF-06 Repreducir música de fondo.....	62
Tabla 33. RQF-07 Reproducir efecto de sonido	63
Tabla 34. RQF-08 Gestionar música de fondo	63
Tabla 35. RQF-09 MenÚ GameOver.....	63
Tabla 36. RQF-10 Reiniciar Nivel	64
Tabla 37. RQF-11 Salir del juego.....	64
Tabla 38. RQF-012 Mover al personaje.....	64
Tabla 39. RQF-13 Saltar.....	64
Tabla 40. RQF-14 Generar plataformas	65

Tabla 41. RQF-15 Destruir plataforma	65
Tabla 42. RQF-16 Actualizar HUD.....	65
Tabla 43. RQF-17 Parallax Scrolling	65
Tabla 44. RQF-18 Guardar puntuación maxima.....	66
Tabla 45. RQF-19 Cargar puntuación maxima	66
Tabla 46. RQF-20 Mover jugador a la derecha	66
Tabla 47. RQF-21 Mover jugador a la izquierda.....	67
Tabla 48. RQF-22 Mover Jugador hacia arriba	67
Tabla 49. RQF-23 Mover jugador abajo.....	67
Tabla 50. RQF-24 Mover jugador diagonal superior derecha	67
Tabla 51. RQF-25 Mover jugador diagonal superior izquierda	68
Tabla 52. RQF-26 Mover jugador diagonal inferior derecha	68
Tabla 53. RQF-27 Mover jugador diagonal inferior izquierda.....	68
Tabla 54. RQF-28 Vida de nuestro personaje	68
Tabla 55. RQF-29 Atacar	69
Tabla 56. RQF-30 Morir.....	69
Tabla 57. RQF-31 Recibir daño.....	69
Tabla 58. RQF-32 Realizar Daño.....	69
Tabla 59. RQF-33 Generar partículas	70
Tabla 60. RQF-34. Destruir particulas.....	70
Tabla 61. RQF-35 Destruir particulas.....	70
Tabla 62. RQF-36. Destruir indicador numérico daño recibido	70
Tabla 63. RQF-37. Generar Indicador numérico daño realizado	71
Tabla 64. RQF-38. Destruir Indicador numérico daño realizado.....	71
Tabla 65. RQF-39. Generar animación cuando nuestro personaje sea alcanzado por un enemigo	71
Tabla 66. RQF-40. Zonas de dialogo	71
Tabla 67. RQF-41. Mostrar zona de diálogo	72
Tabla 68. RQF-40 Cargar nivel	72
Tabla 69. RQF-43. Limite Mapa	72
Tabla 70. RQNF-01 Acceso al almacenamiento del dispositivo	85
Tabla 71. RQNF-02 Almacenamiento mínimo	85
Tabla 72. RQNFR-02 Respuesta Rápida	85
Tabla 73. RQNFE-01 Compatibilidad dispositivos Android	86
Tabla 74. RQNFE-02 Compatibilidad resoluciones dispositivos.....	86
Tabla 75. RQNFI-01 Dispositivo Android	86
Tabla 76. RQNFI-01 Idioma.....	87
Tabla 77. CU-01 Selección juego	89
Tabla 78. CU-02 Jugar Mini-juego	89
Tabla 79. CU-03 Mover personaje Mini-juego	90
Tabla 80. CU-04 Saltar Mini-juego.....	90
T Tabla 81. CU-05 Jugar Modo Desafío	91
Tabla 82. CU-06 Seleccion Nivel	91
Tabla 83. CU-07 Mover personaje Desafío	92
Tabla 84. CU-08 Atacar Desafío	92
Tabla 85. CU-09 Interactuar Desafío.....	93
Tabla 86. CU-10 Menú pausa Desafío.....	93
Tabla 87. CU-11 Reiniciar Nivel Desafío	94
Tabla 88. CU-12 Cerrar menu pausa Desafío	94
Tabla 89. CU-13 Salir a menu selección juego Desafío.....	95
Tabla 90. CU-14 Configuración	95

Tabla 91. CU-15 Salir.....	96
Tabla 92. CU-16 Gestionar música.....	96
Tabla 93. CU-17 Gestión de efectos de sonido.....	97
Tabla 94. Características principales	98
Tabla 95. Ejemplo tabla IU.....	125
Tabla 96.IU-01 HomeScene	126
Tabla 97.IU-02 ConfigurationPanel	126
Tabla 98.IU-03 MainScene	127
Tabla 99.IU-04 HUD Desafío – GameScene	128
Tabla 100.IU-05 GameOver Plataforma	129
Tabla 101. IU-06 SeleccionLvl.....	130
Tabla 102. IU-07 HUD Modo Desafío.....	131
TTTabla 103. IU-08 HUD MenuPausa Desafío	132
Tabla 104. IU-09 MenuGameOver Desafío	133
Tabla 105. IU-10 Ventana Dialogo.....	134
Tabla 106. IU-11 Pantalla Mundo General	135
Tabla 107. IU-12 Pantalla batalla.....	135
Tabla 108. IU-13 Menú Equipo.....	136
Tabla 109. IU-13 Menu Equipo II	136
Tabla 110. PR-01 Abrir Menú Configuración/Ajustes	159
Tabla 111.PR-02 Entrar modo de juego Plataforma	160
Tabla 112.PR-03 Salto modo Plataforma	160
Tabla 113.PR-04 Cargar nivel 1 modo Desafío.....	161
Tabla 114.PR-05 Movimiento+botones personaje modo Desafío	161
Tabla 115.PR-06 Reiniciar Nivel modo Desafío	162

ÍNDICE DE FIGURAS

Figura 1. Ventas mundiales de ordenadores, tablets y smartphones, en millones de dólares, de 2009 a 2014.....	12
Figura 2. Comparativa del volumen de aplicaciones por cada plataforma.....	21
Figura 3. Evolución de los tipos de aplicación de descarga en dispositivos móviles	22
Figura 4. Ciclo de vida de Monobeahvior	38
Figura 5. Arquitectura de Unity3D.....	38
Figura 6. Metodología Waterfall Process.....	39
Figura 7. Árbol de Características.....	99
Figura 8 Arquitectura lógica	105
Figura 9. Arquitectura física	105

ÍNDICE DE DIAGRAMAS

Diagrama 1. Modelo de Casos de Uso	88
Diagrama 2. Diagrama de secuencia para que nuestro personaje salte.....	100
Diagrama 3 Diagrama de secuencia para coger un ítem del juego.....	101
Diagrama 4 Diagrama de secuencia para abrir el menú “pausa” del juego	102
Diagrama 5. Diagrama de secuencia para mover al personaje hacia la derecha(Desafío).....	102
Diagrama 6. Diagrama de secuencia para atacar	103
Diagrama 7. Diagrama de clases de la estructura de Unity3D.....	106
Diagrama 8. Diagrama de clases de los controladores del juego(General).....	107
Diagrama 9. Diagrama de clases del personaje del juego (Desafío)	108
Diagrama 10. Diagrama de clases de los enemigos(Desafío)	108
Diagrama 11. Diagrama de clases de los objetos del modo de juego plataformas	109

Resumen

Este proyecto consiste en desarrollar un videojuego “multi-juego”, es decir, que sean, en este caso, tres juegos en uno, compartiendo historia y complementándose entre ellos.

Modo Plataforma: Juego del género plataformas.

El jugador controlará al personaje principal (Ifrahul) de nuestra historia a lomos de un “chocobi”, montura que le permitirá a nuestro jugador moverse más rápido en el modo Historia/Desafío.

El objetivo será el de recoger los cristales de sellado.

Modo Desafío: Juego del género Acción-RPG

El jugador controla al personaje principal.

El objetivo será el de completar todos los niveles que componen este modo de juego, estos niveles tendrán múltiples posibilidades, desde resolver distintos problemas para completar una determinada misión hasta niveles –modo horda- en la que reaparecerá un número ingente de enemigos.

Modo Historia: Juego del género RPG (No implementado)

El jugador controla al personaje principal de nuestra historia y tendrá que recorrer todo un mundo pasando por misiones, batallas, luchas intelectuales como puzles y zonas secretas...

El objetivo será el de ayudar a nuestro personaje a vengarse del “bien” (del lado de la “luz”) y de los “11” sabios (organismo encargado de mantener el bien y el mal en equilibrio), debido a que estos se aliaron con la intención de sellarlo.

Para la consecución de este objetivo, se llevará a cabo el aprendizaje y uso del motor gráfico Unity3D, debido a su gran potencial.

Palabras clave: videojuego, 2D, Unity3D, Android, dispositivo móvil, Multi-Juego.

Abstract

This Project is about developing a “multi-game” videogame where three games will complement each other in order to create a story.

Plataform Mode: Game of the platforms genre.

The player will control the main character (“Ifrahul”) of our story on the back of a chocobi. The riding saddle will allow our player to move faster in the challenge.

The main objective will be to collect the sealing crystals.

Challenge Mode: Genre Game Action-RPG

The player controls the main character.

The main aim will be to complete all the levels that make up this game mode. Those levels will provide the user with a wide range of possibilities, from different problems that will need to be solved in order to complete a certain mission to levels – Horde Mode- in which a great number of enemies will reappear.

Campaign/Story mode: RPG Genre Game (Not implemented)

The player controls the main character of our story and will have to overcome different quests, win battles or intellectual fights such as puzzles to be able to travel around the world.

The objective will be to help our character to take revenge on the "good" (on the side of the "light") and on the 11 wise men (organism in charge of keeping good and evil in balance) because they allied themselves with the Intend to seal him.

For this Project, i will use the Unity3D graphics engine due to its great potential.

Keywords: videogame, 2D, Unity3D, Android, mobile device, Multi-Game

Capítulo 1

Introducción

El objetivo principal de este primer capítulo, es proporcionar al lector una visión general del proyecto. Presentando el problema a tratar, como solucionarlo, que motivación a llevado a su realización, los objetivos a conseguir y la puesta en práctica con las herramientas utilizadas para satisfacer las anteriores cuestiones.

Este documento contiene toda la información relativa al proyecto “**Desarrollo de un videojuego en un entorno móvil con Unity3D [The dark World-The Age Of Darkness]**”, basado en la creación de un prototipo básico de videojuego para dispositivos móviles Android cuyo diseño se caracterizará por la utilización de gráficos 2D, la implementación se llevará a cabo con la ayuda del motor gráfico Unity3D.

El juego se basa en un estilo “multi-juego”, y van ha estar basados en la captación de influencias de juegos “retro”. Los tres modos de juegos serán los siguientes.

Juego género plataformas: Este género se caracteriza por tener que controlar a un personaje, el cual tendrá que correr y saltar por un mapa con un desplazamiento horizontal ayudándose de una serie de plataformas mientras se intenta recoger todos los objetos posibles del nivel.

Juego género Action-RPG: Este género se caracteriza por compartir muchas características con el próximo género (RPG), la diferencia de estos es que un Action-RPG se centra en combates en tiempo real, en los que el jugador se puede desplazar a lo largo del mapa tanto de manera vertical como de manera horizontal mientras ataca a sus enemigos.

Juego género RPG: Este género consiste se caracterizará por tener combates por turnos.
(No implementado en esta versión del proyecto)

En este TFG se implementará un videojuego en 2D que permitirá hacer disfrutar al usuario/jugador, permitiéndole optar a la selección de distintos tipos de juego. A su vez, para mejorar el acoplamiento entre los tres modos, serán juegos con características comunes y los cuales podrán interactuar entre ellos, ya sea en el desbloqueo de algún objetivo secundario, así como mera información para completar la información de la historia.

Sin embargo, hay que tener en cuenta que a la hora de desarrollar un videojuego, no existen tutoriales o manuales de cómo abordar el desarrollo con una metodología concreta, y, se requieren conocimientos de diferentes disciplinas y herramientas. Esto, se refleja en equipos conformados por distintos equipos con un alcance temporal amplio.

1.1 Motivación

En esta sección se describirá brevemente las razones que han influido en el autor del proyecto para la realización del mismo.

La motivación principal que ha ayudado en la realización del proyecto es el interés que han ejercido los juegos desde que era pequeño, esto ha llevado a querer aumentar mis conocimientos en el área de los videojuegos, con el objetivo de en un futuro trabajar en dicho sector, y el interés de pasar del punto de vista “Jugador” al de desarrollador de videojuegos, introduciéndonos más en esta área.

Además, a todo esto, se le une que son aplicaciones intergeneracionales, es decir, que son utilizados tanto como por jóvenes como por personas mayores indistintamente de su edad, lo que conlleva un aumento en el alcance de las personas a las que podrán llegar.

Todo esto se puede aplicar a los distintos géneros de videojuegos, así como las videoconsolas que los soportan (portátiles, no portátiles), debido a que las videoconsolas portátil son de las mas utilizadas actualmente (dispositivos móviles) el juego irá enfocado en este tipo de dispositivo, ya que dan mas comodidad al usuario al poder llevarse el juego a cualquier parte, y como no será necesario Internet, se podrá jugar en cualquier lugar simplemente con tenerlo instalado.

A continuación se muestra un gráfico en el que se puede observar que ya en 2014 la venta de dispositivos móviles superaba en gran medida a tecnologías como ordenadores y tablets.

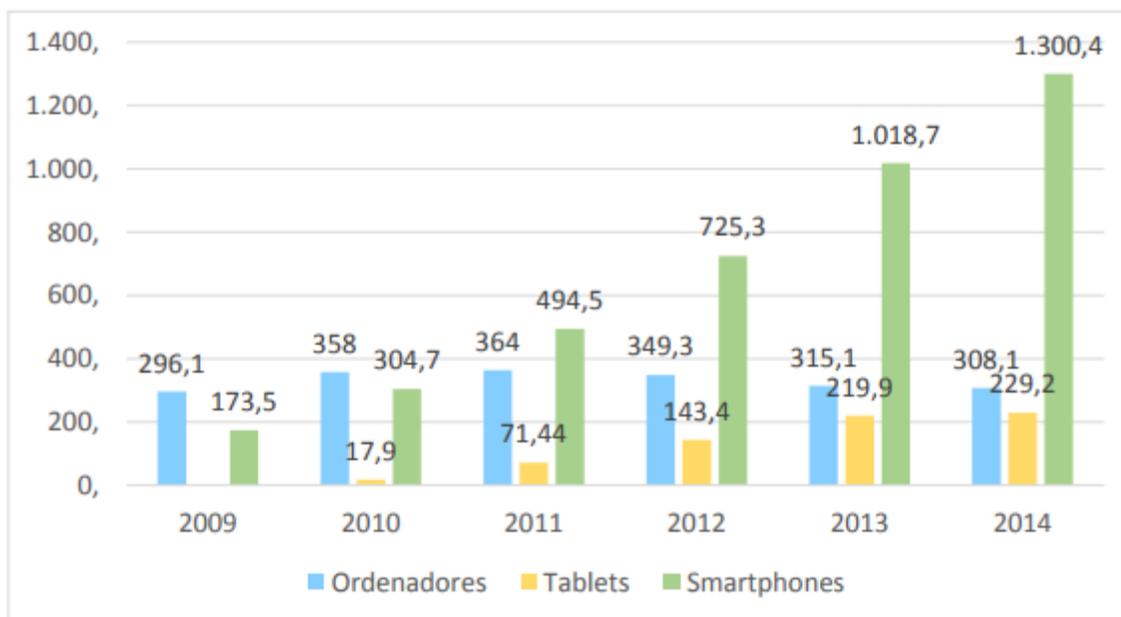


Figura 1. Ventas mundiales de ordenadores, tablets y smartphones, en millones de dólares, de 2009 a 2014

El mundo de los videojuegos avanza sin parar, y, actualmente, es una de las mayores industrias del mundo, aumentando cada día debido a que hay más y mejores herramientas que permiten a pequeños desarrolladores crear videojuegos y lanzarlos en distintas plataformas, con menores dificultades.

Por otra parte la evolución de los teléfonos y dispositivos móviles ha dado paso a los smartphones y tablets, entre otros dispositivos, que actualmente poseemos. Estos dispositivos ya han pasado a formar parte de la vida diaria de la mayoría de las personas, llegando a ser imprescindibles en muchos casos.

El mercado de los videojuegos no ha parado de crecer desde su origen, demostrando que aun tratándose de un producto de entretenimiento, es uno de los más valorados por los consumidores y actualmente se refleja fácilmente en la cantidad de torneos y sus cuantiosas recompensas.

Con la aparición de los juegos “indies” o independientes, el mercado de los videojuegos ha crecido de manera exponencial debido a que, para crear un videojuego de este estilo, no es necesario disponer de un gran equipo, ni presupuesto sino que están caracterizados principalmente por su simplicidad. (Se estima que el 90% de las cuatrocientas empresas españolas relacionadas con el mundo del videojuego se declaran independientes.)

Más allá de las grandes producciones, llamados “triple A”, como podrían ser <<Battlefield>>, <<Grand Theft Auto>>, títulos mundialmente conocidos y con grandes equipos, comienzan a entrar en acción creadores que, con creatividad y buenas ideas, intentan abrirse hueco en este sector, apostando por los juegos “indies”.

Algún ejemplo de éxito en esta industria podrían ser: AngryBirds o el famoso Flappy Bird, un juego sencillo, el cual su autor decidió retirarlo debido a que consideró que era demasiado adictivo, ganando solo en publicidad 50.000 dólares diarios.

Por todo ello, se ha decidido para este proyecto crear un videojuego que rompa con las estéticas de las grandes compañías, que, aun con un nivel de detalle en gráficos simple sea atractivo para el usuario, no sólo por su jugabilidad sino por su historia y disfrute, es decir, volver a traer los juegos de antaño, que aun teniendo una capacidad de unos cuantos megas la diversión proporcionada no era comparable.

La mayor particularidad de este proyecto en lo relativo a los modos de juego a implementar será que, romperá con la estética de “*Los Buenos tienen que salvar al mundo de los malos*” se quiere dar a entender con este videojuego, que ni los “malos” son tan malos siempre, ni los “buenos” son tan buenos siempre, otorgándole carácter de reflexión por parte del usuario, por lo que el jugador tendrá que controlar a “Ifrahul” el cuál sera el personaje principal de nuestro juego y, llevarle a la victoria en los tres modos de juego¹.

¹ El modo plataforma no tendrá fin.

1.2 Objetivos

En esta sección se describirá brevemente los objetivos y sub-objetivos principales en la realización del TFG:

El objetivo de TFG es el desarrollo de una “demo” de un videojuego en 2D desarrollado con unity3D, esté se realizará desde cero, por lo que se mostrará todo el proceso de desarrollo, hasta conseguir la creación de múltiples videjuegos dentro del videojuego principal y cada juego con sus características propias.

Para conseguir este objetivo, deben cumplirse los sub-objetivos generales siguientes:

- **Diseño del videojuego:** personaje principal del juego, compañeros/aliados de nuestro personaje, enemigos, escenario por el que mover al personaje, objetos a recoger y todo tipo de objetos decorativos para el mapa del mundo...
- **Creación de una Interfaz de Usuario Gráfica** sencilla, llamativa y con temática “oscura”.
- **Creación del sonido del juego.** Lo ideal será la creación de distintos tipos de música para cada modo de juego y, además, la temática también sera de tipo “oscuro”.
- **Programación de las funcionalidades básicas del videojuego.**
- **Creación de Inteligencia Artificial** para los enemigos.
- **Creación de un nivel infinito** → Modo **plataforma..**
- **Creación de distintos niveles** → Modo **desafío.**
- **Creacion de un mundo extenso** por el que el personaje debiera ir avanzando en su historia → Modo **campaña.** (No implementado)

Con todo ello se pretende aprender a utilizar **Unity3D**, y adquirir la capacidad de desarrollar un videojuego desde cero, comprendiendo su proceso y requerimientos, poniendonos en la práctica, en el lado de un desarrollador de videojuegos y comprendiendo los retos que han tenido que superar juegos exitosos o, simplemente que fueran del agrado del autor de este proyecto.

Otros sub-objetivos secundarios que se llevarán a cabo como “hobbie” trás la presentación de una primera “demo” con las funciones básicas serán:

- **Permitir al usuario guardar el progreso del juego** para seguir en el punto donde lo había dejado, ya sea en modo desafío o en modo campaña.
- **Desarrollar un modo campaña**, el cual nos contará la mayor proporción de la historia de nuestro personaje (debido a que la historia se contará a través de los 3 modos de juego, cabe destacar que se espera que el 75-80% de la historia como tal se encontrará en el modo historia).

1.3 Alcance

En esta sección se describirá brevemente a quiénes se quiere llegar con el desarrollo del proyecto.

El juego va dirigido a cualquier persona que disponga de un dispositivo móvil Android, el único requisito será el de tener tiempo libre y ganas de disfrutar con una buena historia que nos tenga enganchados a la par que el videojuego.

La dificultad del videojuego dependerá del modo de juego al que el usuario quiera enfrentarse.

-Platform(Mini-juego): Dificultad apta para cualquier persona.

-Action-RPG(Desafío):Dificultad apta para cualquier persona pero se centrará más en aquellas que les interese más la consecución de niveles con cierta dificultad, ya sea de lógica o simplemente superar niveles “horda” en los cuales saldrán numerosos enemigos a los que tendremos que vencer para así completar dicho nivel.

-RPG(Campaña): Dificultad apta para cualquier persona, aunque para determinadas misiones habrá que estar muy atento...

1.4 Herramientas Utilizadas

En esta sección se describirá brevemente las herramientas a utilizar para el desarrollo del TFG:

Para el desarrollo del TFG se va a hacer uso de las siguientes herramientas:

- **Unity3D**: Unity3D es un motor gráfico para crear videojuegos desarrollado por Unity Technologies. Es un motor muy potente el cual permite la creación de juegos multiplataforma (2D/3D) además, dispone para desarrolladores inexperimentados (y experimentados) una gran cantidad de información de ayuda.

- **MonoDevelop:** IDE utilizado por Unity3D.
- **C#:** También conocido como “*C-Sharp*”, se trata de un lenguaje de programación “C” orientado a objetos. Se ha decidido la utilización de dicho lenguaje debido a que la mayor parte de personas de la comunidad de unity lo utilizan, lo que facilita la resolución de dudas, sobre todo para alguien que empieza de cero con esta tecnología.
- **Gimp:** Es la única herramienta utilizada para el desarrollo de los diseños de todos nuestros elementos gráficos del juego, spriters (mapas de bits en 2D), estos han sido creados en su totalidad pixel a pixel.

Las animaciones han sido realizadas “frame by frame”, lo que ha resultado un trabajo un tanto tedioso pero con una finalización satisfactoria.

- **Android SDK:** Kit de Desarrollo Software de Android.
- **<http://www.superflashbros.net/as3sfxr/>:** página web online que permite la creación de efectos de sonido mezclando distintos sonidos de una manera sencilla.
- **Microsoft word 2007:** Procesador de texto.
- **StarUML, Dia:** Software para la realización de diagramas.
- **Draw.io:** Herramienta online de creación de diagramas
- **FL Studio:** Aplicación para crear/modificar música.
- **UnityRemote:** Se trata de una aplicación móvil que nos permitiera ejecutar el Editor de Unity desde el dispositivo de manera remota, es decir, sin necesidad de exportar o instalar el juego. Esta aplicación ha sido utilizada en los últimos retoques del videojuego debido a que no se tenía constancia antes de ella, pero para el futuro desarrollo del juego se utilizará con mayor frecuencia,

1.4.1 ¿Por qué Unity3D?

En esta sección se describirá brevemente el por qué del uso de Unity3D para la realización del proyecto.

Se ha decidido hacer el uso de Unity3D debido a que actualmente es uno de las plataformas de desarrollo de videojuego más potentes y, permite una fácil integración multiplataforma permitiendo desarrollar videojuegos para ordenador, videoconsola, dispositivos mó-

viles...

Unity es un motor de videojuego multiplataforma creado por Unity Technologies, esté, inicialmente se trataba de un entorno de desarrollo de juegos 3D, pero con el paso del tiempo y las versiones, ya en su versión 4.6 se integró en él la posibilidad de desarrollar en 2D, haciendo frente a motores como Cocos2D o Marmalade, los cuales estaban diseñados para este fin.

El éxito de Unity se debe en parte a que han sabido enfocarlo para poder satisfacer las necesidades de desarrolladores independientes, desarrolladores los cuales tienen difícil crear su propio motor de juego o adquirir licencias.

El enfoque de Unity Technologies en este área es la de “*democratizar el desarrollo de los juegos*”, y contenidos interactivos en 2D/3D, es decir, conseguir hacer más accesible este contenido a cuantas más personas sea posible.

Una de sus ventajas principales, es que debido a lo anteriormente comentado, es gratuito, aunque tiene limitaciones, como la aparición obligatoria de su logo en el “SplashScreen” (pantalla previa al juego) o el no poder utilizar su potencial al 100% pero, sin embargo, con todos los “planes”(gratuito-de pago) se puede tener acceso a todas las prestaciones del motor, desarrollar para múltiples plataformas, acceso a todas las actualizaciones y no es necesario pagarles “regalías” por cada juego diseñado y lanzado al Mercado.

La mayor ventaja que nos encontraríamos, sería su extensa documentación y comunidad lo cual ayudará a comprender algo más fácil este “mundo”.

Además, se caracteriza por centrarse en las funcionalidades del juego y no en la gestión de memoria, punteros... ahorrando gran cantidad de tiempo en el desarrollo de videojuegos

Por todo lo comentado anteriormente, actualmente, Unity3D una de las plataformas más utilizadas y potentes en esta área.

1.4.1 ¿Por qué Android?

En esta sección se describirá brevemente por qué se ha destinado el proyecto al entorno Android.

Principalmente se ha elegido Android debido a las siguientes ventajas:

- **Código libre:** Posibilidad de modificarlo y compartirlo libremente.
- **El más utilizado actualmente:** Actualmente el SO más distribuido son dispositivos que poseen Android, debido a su gran potencial.

- **Precio:** Esta característica, influye en que sea más utilizado. Debido a que la mayoría de compañías disponen de móviles con dicho sistema operativo, se pueden encontrar dispositivos muy baratos con gran relación calidad/precio.
- **Google:** Debido a su adquisición por parte de google (en 2005), permite disponer de cantidad de servicios de esta compañía, algunos los cuales actualmente están muy unidos a nuestra vida cotidiana como puede ser “YouTube”.
- **Multitarea:** Permite la ejecución de distintas aplicaciones (.apk’s) a la vez, si necesidad de cerrarlas, permitiendo ponerlas en Segundo plano.
- **GooglePlay:** Una de las ventajas es que las aplicaciones de la tienda de android, en su mayoría, son gratuitas. Además, los desarrolladores deberán pagar solamente una licencia a la hora de registrarse en la tienda, a diferencia de la AppleStore que se deberá pagar cada vez que se desee publicar una aplicación en la tienda.
- **Exportación:** Unity3D permite exportar de manera sencilla nuestros proyectos a Android de manera gratuita, algo que no pasaría con iOS.
- **Portabilidad:** Es un dispositivo el cual nos acompaña actualmente a todas partes y se considera una gran ventaja el poder hacer uso de él en cualquier momento.

Aunque también se podrían encontrar algunas desventajas como:

- **Seguridad:** Al permitir tanta libertad a la hora de publicar aplicaciones hay una mayor probabilidad de poder meter aplicaciones maliciosas, cabe destacar que Android dispone de varios controles de seguridad para intentar evitarlo.
- **Gasto de batería:** El sistema operativo Android consume mucha batería debido a su característica multitarea.

1.5 Organización del documento

En esta sección se describirá brevemente la estructura y el contenido de cada parte del documento.

Se dividirá en Capítulos, cada capítulo con distintas secciones:

- **Capítulo 1. Introducción:** En este capítulo, se presenta el contexto que engloba al análisis y al desarrollo del proyecto.
Se encontrarán descritos los puntos más representativos del proyecto así como las ideas previas al desarrollo, las motivaciones para el mismo y se expondrán los objetivos que se deasean alcanzar a lo largo del proyecto.
- **Capítulo 2. Estado del Arte:** En este capítulo se mostrará una visión global sobre la temática de los videojuegos, analizando la situación actual del entorno de desarrollo y juegos similares.
- **Capítulo 3. Introducción a Unity3D:** En este capítulo encontramos una breve introducción al motor gráfico Unity3D, una breve explicación de su estructura y capacidades así del porqué de su utilización y de su modo de uso.
- **Capítulo 4. Planificación y Presupuesto:** En este capítulo se establece la metodología a seguir. La planificación y el presupuesto del proyecto.
- **Capítulo 5. Análisis:** En este capítulo se expone el análisis previo al desarrollo del videojuegos, identificando los requisitos. Casos de Uso. Diagramas de secuencia. Árbol de Características.
- **Capítulo 6. Diseño:** En este capítulo se encuentra el diagrama de clases de diseño. Arquitectura software. Diagramas de Estados. Diseño de Interfaz. Diseño del juego.
- **Capítulo 7. Implementación:** En este capítulo se explica la parte del desarrollo del proyecto en su fase de implementación.
- **Capítulo 8. Pruebas:** En este capítulo se encuentran las pruebas realizadas durante el desarrollo del videojuego.
- **Capítulo 9. Manual de Instalación:** En este capítulo se explican las posibles formas de instalación del videojuego.
- **Capítulo 10. Manual de Usuario:** En este capítulo se explica el modo de uso del videojuego.
- **Capítulo 11. Conclusiones:** En este capítulo se encuentran las conclusiones tras el desarrollo del proyecto.

- **Capítulo 12. Bibliografía:** En este capítulo se encuentran las referencias y la bibliografía utilizada a lo largo del proyecto para el desarrollo del mismo.
- **Capítulo 13. Glosario:** vocabulario técnico, argot, siglas, contenidas en el proyecto.
- **Capítulo 14. Contenido del CD-ROM.**

Capítulo 2

Estado del arte

En este capítulo, se trata de analizar la situación actual de nuestro videojuego en el entorno en el que se desenvuelve, realizando una comparativa con las opciones actuales.

Se ha decidido dirigir el tema del proyecto hacia el área de los videojuegos ya que, estos llevan con nosotros desde hace décadas y su evolución en estos últimos años lleva un ritmo vertiginoso, convirtiéndose en una de las mayores industrias del mundo. Además, irá dirigido a dispositivos móviles debido a que es el dispositivo más utilizado en la actualidad, superando al número de personas en el mundo, las cuales, cada día son más dependientes.

La plataforma elegida será “GooglePlay”, como vemos en el grafico siguiente, ha ido cogiendo fuerza estos ultimos años en el desarrollo de aplicaciones y ya en 2015, esta, obtuvo el doble de descargas que la AppStore de Apple, ya que, la mayor parte de la gente dispone de un móvil Android.

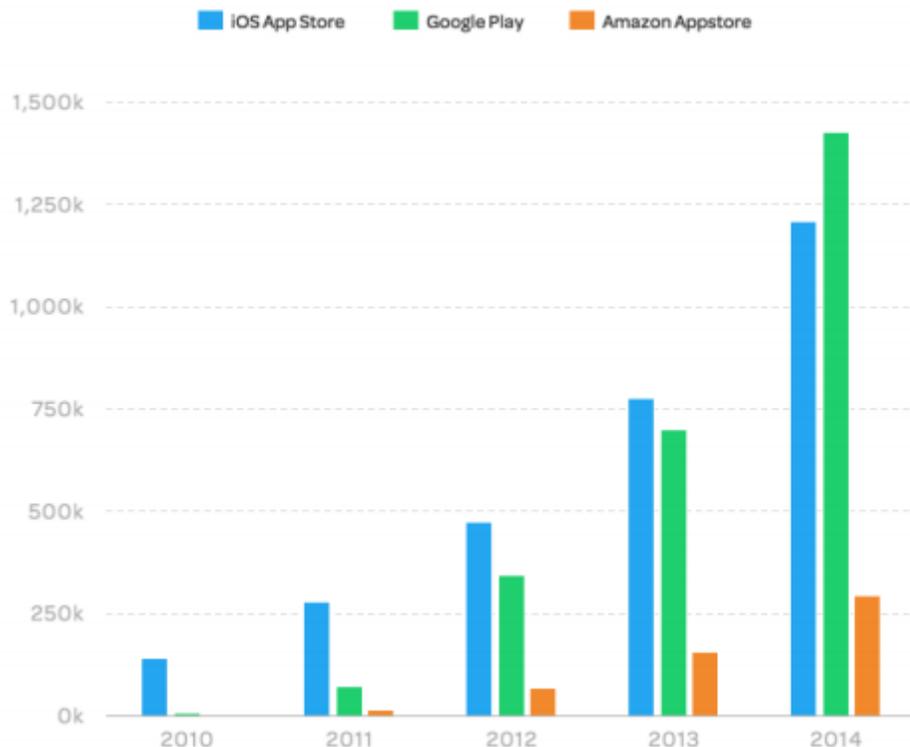


Figura 2. Comparativa del volumen de aplicaciones por cada plataforma.

De esta diferencia, se puede apreciar que la mayor parte de los beneficios obtenidos por esta plataforma son ingresos obtenidos de descargas de videojuegos (Aproximadamente 90%). Por ello, el videojuego irá dirigido al GooglePlay.

Podemos ver como la videojuegos han ido ganando importancia a lo largo de los años, llegando a ser el principal tipo aplicación descargada por los usuarios.

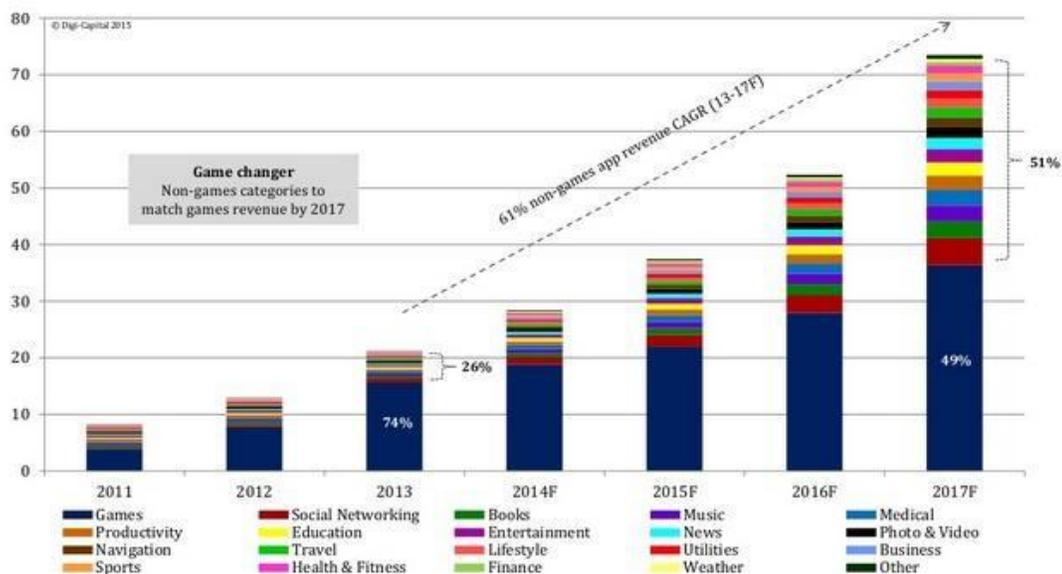


Figura 3. Evolución de los tipos de aplicación de descarga en dispositivos móviles

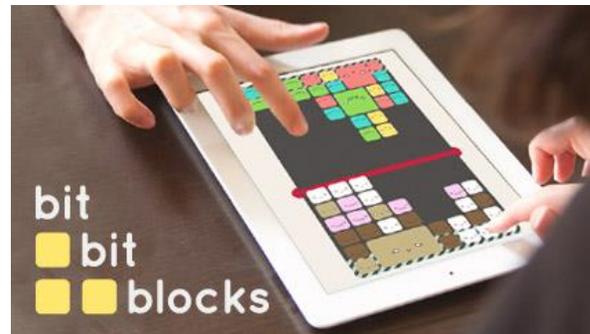
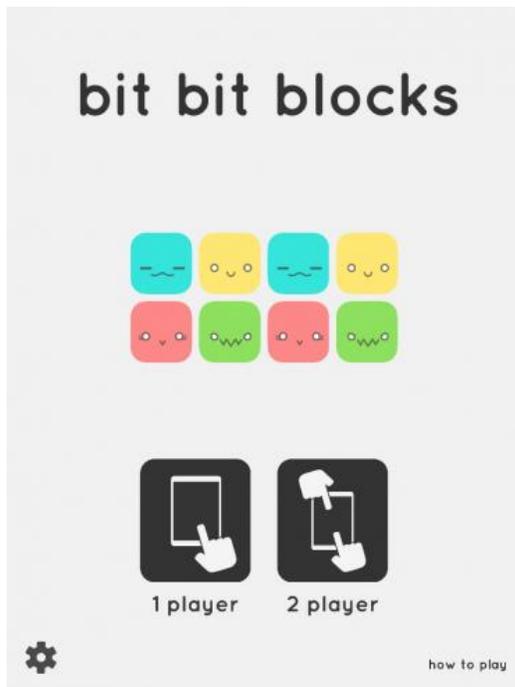
Tras analizar los datos, y, debido a que uno de los objetivos principales del juego es que sea portable para así aumentar las hora de diversión, entre todas las opciones multiplataforma que nos ofrece Unity, sea ha decidido desarrollar el videojuego para Android

En este caso, el desarrollo de proyecto se llevará a cabo con las características parecidas a lo que se denominan juegos “indies” los cuales son producidos por pequeños equipos, que sin el apoyo económico de las grandes compañías, desarrollan videojuegos que prescindan de grandes gráficos e historia, centrándose en la esencia del juego, haciéndolo divertido y no repetitivo con la diferencia de que todo será realizado por el autor de este proyecto.

Como vemos, estas características describen a grandes rasgos este proyecto, ya que consiste en una única persona encargada del diseño de los gráficos, programación, música y argumento del videojuego, con un presupuesto bajo pero por contra se centrará en la historia intentando hacer el videojuego lo más divertido y entretenido posible.

Un ejemplo de este tipo de juegos “Indies” podría ser:

- **Bit bit blocks**: juego de puzzle competitivo muy vistoso cuya mecánica es simple, unir cuadrados del mismo color para lanzárselos a tu rival.



Juego “Indie” bit bit blocks

Llegó a ser uno de los tres mejores juegos “indies” para Android según Google en 2016.

2.1 Juegos de referencia

En esta sección se describirán brevemente los juegos que han influido a la hora de la realización de este proyecto y, su actual estado en el Mercado.

Estos juegos estarán divididos en tres partes las cuales se referirán a cada modo de juego disponible y se hablara tanto de las referencias que nos han llevado a la realización de este proyecto como el estado actual del Mercado.

Modo Plataforma (Mini-Juego)

1. Referencia Pasada



Sonic

Este videojuego de plataformas fue, durante mucho tiempo, considerado el buque insignia de Sega, convirtiéndose en uno de los juegos más vendidos de la consola con 15 millones de copias vendidas.

Se caracteriza por llamarse igual que su personaje principal (“erizo azul”), y cabe destacar que en un principio, según varios comic (Saga compuesta por juegos, comics, series...) esté era marron pero al alcanzar la verlocidad del sonido se vuelve azul.

La misión de Sonic en este juego de la saga es la de salvar a los animales y las “chaos Esmelards” de su Isla, isla a la que “Robotnik”, su enemigo, habia conquistado y cuyo objetivo era la de esclavizar a los animales que residian en esa isla atrapandolos dentro de robots.

El fuerte de este juego es su jugabilidad e historia, historia que no acaba en este juego, sino que hay toda una saga por descubrir...



✓ **Pros:** controles, fluidez, checkpoints, historia, jugabilidad.

✗ **Contras:** gráficos (contando para la época actual 2017), duración.

2. Referencia Actual

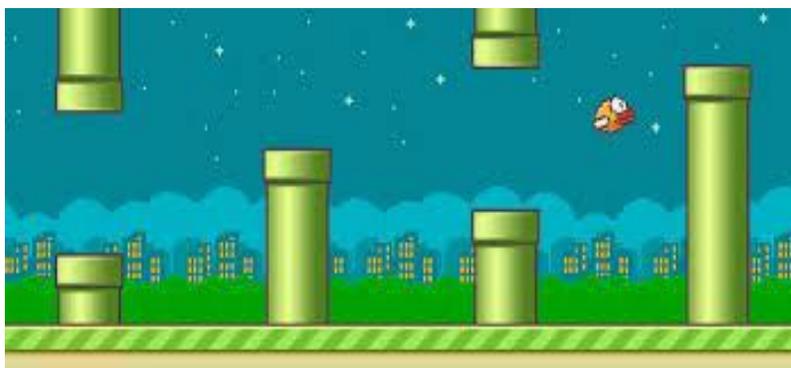


Flappy Bird

Flappy Bird es un juego para móviles desarrollado en 2013 y publicado por .GEARS Studios, un pequeño desarrollador de juegos independiente.

Como ya se explicó anteriormente, debido a la adicción a él, el juego, publicado el 24 de mayo de 2013, fue eliminado de App Store y Google Play por su creador el 9 de febrero de 2014.

El juego consistía en controlar un pájaro intentando volar entre filas de tuberías verdes sin tocarse con estas. . El desarrollador creó el juego en varios días utilizando como personaje central un pájaro que diseñó para otro juego cancelado en 2012.



✓ **Pros:** controles, fluidez, duración.

✗ **Contras:** gráficos, historia.

Modo Action-RPG(Desafío)

1. Referencia Pasada



Zelda- The minish cap

Este juego es el duodécimo videojuego de la saga The Legend Of Zelda, desarrollado por Flagship bajo supervisión de Nintendo.

The minish cap es el primer título original de la saga para consola portátil (GBA- Game Boy Advance) y llegó a la gente en el noviembre del 2004, excepto para la gente de Estados Unidos y Australia que fue a principios del 2005.

The minish cap es el tercer título de The Legend Of Zelda y conserva elementos comunes de anteriores ediciones de la saga, como el lugar en el desarrollo de la historia, aparecen miembros de “razas” anteriormente descritas y ha esto, se le añaden nuevos elementos. Esto, sin lugar a duda, es uno de los mayores Fuertes de este juego, su “GRAN” historia.

The minish cap fue calificado como el vigésimo mayor juego de Game Boy Advance según IGN²² y nombrado "El Juego del Año" en 2005 por GameSpot.



²² IGN: Sitio web que principalmente se centra en noticias sobre videojuegos.

✓ **Pros:** Historia, jugabilidad, checkpoints, sencillez, duración.

✗ **Contras:** gráficos (contando para la época actual 2017)

2. Referencia Actual

El juego más parecido a mi ideal como modo de juego que he encontrado en el google play ha sido Bit Dungeon.



bit Dungeon se trata de un juego cuyo género es de acción roguelike con un mundo gigante para explorar.

El personaje se trataría de un espíritu en un mundo de demonios y debe luchar para conseguir la paz a su alma.



✓ **Pros:**

✗ **Contras:** juego de pago.

Debido a que el juego es de pago no se ha podido probar y no se podrían sacar con buen criterio los pros y contras del juego.

Modo RPG(Campaña/historia) (No implementado)

1. Referencia Pasada



Se trata de la sexta entrega de la saga Final Fantasy. El primer juego fue lanzado al mercado para la Super Nintendo en 1994.

El argumento del juego gira alrededor de un grupo, denominado Los replicantes, que lucha contra la tiranía del Imperio, un régimen militarista de inspiración bonapartiana gobernado de forma despótica por el Emperador Gestahl.

Esta considerado un hito sin precedentes dentro de la saga y del genero RPG.



✓ **Pros:** Historia, jugabilidad, checkpoints, duración.

✗ **Contras:**

2. Referencia Actual

Actualmente también podemos encontrar diversas versiones remasterizadas del “Final Fantasy”, la contra es que la versión más barata vale 6,99\$

Otro sería un juego llamado “Chrono Trigger” aunque también sería un juego de pago. Debido a que son los únicos que se asemejan al estilo de juego que se quiere conseguir se hablara algo de este juego por no volver a repetir juego.



Chronos Trigger se trata de un desarrollado y publicado por Square para la videoconsola Super Nintendo Entertainment System (SNES). Es el primero de los títulos que componen la saga Chrono.

La historia del juego sigue a un grupo de jóvenes aventureros que fueron transportados a través del tiempo accidentalmente y descubrieron que el mundo iba a ser destruido en un futuro lejano. Jurando prevenir este desastre, viajan a través de distintas épocas para descubrir la forma de salvar al planeta.



✓ Pros:

✗ Contras: juego de pago.

Debido a que el juego es de pago no se ha podido probar y no se podrían sacar con buen criterio los pros y contras del juego.

Capítulo 3

Introducción a Unity3D

En este capítulo se llevaran a cabo:

1. Introducción al motor gráfico Unity3D (información General +Características)
2. ¿Por qué finalmente se ha decidido utilizar Unity3D para este proyecto?
3. Modo de uso de Unity3D.

1. Introducción al motor gráfico Unity3D

Unity es un motor gráfico para crear videojuegos desarrollado por Unity Technologies.

Inicialmente, unity3D solamente se diseñó para el desarrollo de videojuego 3D, pero con el paso del tiempo y las versiones, ya en su versión 4.6 se integró en él la posibilidad de desarrollar en 2D, haciendo frente a motores como Cocos2D o Marmalade, los cuales estaban diseñados para este fin. (Exceptuando planes de pago en Unity, esté, sigue siendo un motor más potentes que los anteriormente nombrados)

Las plataformas para poder desarrollar con Unity3D son: Windows, OS X y Linux.

Los juegos desarrollados con este motor tienen la característica de ser juegos multi-plataforma, con pequeños cambios en los controles se puede gestionar fácilmente su cambio, algunas plataformas a las que podrían ir destinados los juegos son:

- **PC de Sobremesa:** PlayStation 4, Xbox ONE...
- **Dispositivos móviles:** Android, IOS...
- **SmartTV:** AndroidTV, tvOS...
- **Realidad Virtual:** Oculus Rift, PlayStation VR...
- **Web:** con WebGL

Pero, sin lugar a duda, una de las mayores ventajas que posee Unity es el gran contenido informativo destinado a todos los usuarios (ya sean primerizos o expertos en el uso de este motor), así como su amplia comunidad que permite solucionar dudas de manera mas rápida.

Esta amplia documentación informativa es proporcionada a través de su web³, y, además, se muestra de forma multilingüe, para que independientemente del idioma del usuario sea factible entender toda la documentación, tanto en javascript como en C#(C-Sharp)



Panel Ayuda General Web Unity

Esta documentación viene complementada con tutoriales, proporcionado también por Unity, lo cual facilita el aprendizaje para los primerizos en esta tecnología. Cuenta con una amplia comunidad, lo que lleva a ayudar a aquellos, ya sean primerizos o no, que tengan cualquier tipo de duda mediante el post de dudas en foros y su posterior respuesta por parte de está.

Por último, Unity tiene su propia <Tienda> de assets, tanto gratuitos como de pago, con esto nos referimos a paquetes de recursos ya desarrollados, desde recursos para llevar a cabo una funcionalidad de un juego hasta podemos encontrar juegos completos. Esto ayuda a cualquier programador ha ahorrar tiempo ya que no hay la necesidad de programarlos uno mismo. (Su acceso puede llegar a realizarse desde la propia plataforma de Unity3D)

La mayor desventaja sería que no es 100% gratuito, aunque no es totalmente de pago, podemos encontrar que Unity nos ofrece distintos planes dependiendo de para que uso lo queramos o los ingresos que produzca nuestro juego.



Planes Unity3D

³ <https://docs.unity3d.com/Manual/index.html>

Elijamos el plan que elijamos tendremos a nuestra disposición gran cantidad de posibilidades en el desarrollo de videojuegos.

Con todas se puede tener acceso a todas las prestaciones del motor, desarrollar para múltiples plataformas, acceso a todas las actualizaciones y no es necesario pagarles “regalías” por cada juego diseñado y lanzado al mercado.

Sin embargo, existen diferencias notables, aunque sean pequeños detalles como podrían ser:

- Con la versión **personal** no es posible la elección de un <SplashScreen>, el cual es la imagen inicial al comienzo del videojuego, en este caso, será obligatorio el uso del logo de Unity, con los demás planes se podrá elegir uno a elección propia o bien inhabilitarlo.

- Los ingresos, tienen un límite por plan, al llegar a cierto umbral deberíamos pasarnos al siguiente plan:
 - ◆ Personal: Límite máximo producido por juego de 100.000\$

 - ◆ Plus: Límite máximo producido por juego de 200.000\$

 - ◆ Pro: Sin límite de ingresos.

 - ◆ Enterprise: Sin límite de ingresos.

2. ¿Por qué se ha decidido el uso de Unity3D?

En principio, se estuvo barajando la opción de realizar el juego mediante COCOS2D, pero finalmente se optó por Unity3D debido a que todo el aprendizaje se ha llevado de forma autónoma y como anteriormente se ha expuesto, Unity3D, incluso en su versión más básica, ofrece un gran potencial para el desarrollo de videojuegos, siendo estos acompañados por una gran documentación, comunidad, tutoriales y la facilidad de exportar a multiplataforma con la única necesidad de realizar los cambios oportunos en lo relativo a con qué interactuemos con el juego (Mando, teclado, pantalla táctil...), específicos de cada “consola”.

3. ¿Modo de uso/Primeros pasos?

Los proyectos e unity3D estan compuestos por distintos Assets (paquetes de recursos), y estos son los que compondran nuestro proyecto.



Interfaz de Unity3D

Unity3D posee una interfaz “amigable”, a la que es fácil acostumbrarse, si no fuera suficiente, se pueden recolocar todas las ventanas que componen su interfaz para dejar la configuración con la que más a gusto nos sintamos y guardar dicha configuración en forma de Layout por posibles futuros cambios

Podemos dividir la interfaz en distintas partes, ordenadas en este caso, según nos vamos familiarizando con Unity

- 1. Project:** Aquí encontraremos la estructura de nuestro proyecto, para facilitar la búsqueda de recursos se aconseja el subdividirla en sus distintos componentes (Sprites,Scripts,Animations,Fonts,prefabs,scenes,StreaminAssets...).
- 2. Hierarchy:** Una vez abierta una escena de un proyecto en Unity, nos mostrará todos los objetos de los que la escena actual hace uso, así como los padres y los hijos de estos, ofreciendo una forma fácil de acceder a ellos y poder modificarlos
- 3. Inspector:** Informe detallado de todas las características del objeto seleccionado, es una parte fundamental en la interfaz ya que en esta ventana se van a llevar a cabo las modificaciones de los objetos, desde la puesta en unity (modificar escalado, editar sprites, cambiar propiedades de la imagen...) hasta su posterior uso (modificación de sus atributos, incluir scripts en dichos objetos...)

4. **Scene:** Cada una de las pantallas que compondrán el juego.
5. **Game:** Nos muestra cómo se va a ir viendo el juego para las distintas resoluciones de pantalla
6. **Console:** Nos ayudará mostrándonos mensajes de error cuando se cometa algún fallo, ya sea en programación o mediante la interfaz gráfica, y, además, nos ayudará mostrándonos mensajes de debug que pongamos en nuestro proyecto.
7. **Animation:** Puede ser utilizada para pre-visualizar y editar los “clips de animación”.
8. **Game controller:** Se encuentran los botones para interactuar con el juego(ventana -Game-).

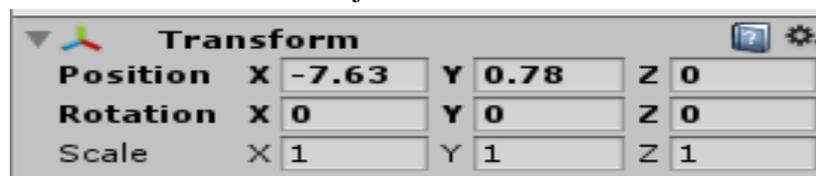
- i. **Play:** Ejecuta/detiene el juego.
- ii. **Pause:** Pausa el juego.

En cuanto a las características del editor para este proyecto, debido a que se tratará de un juego en 2D, todos los materiales y físicas estarán adaptados al 2D.

Utilizaremos los Sprites, mapas de bits 2D, para la realización de todo el contenido visual de la aplicación, ya sea el personaje principal, elementos decorativos, mapas, interfaz de usuario...

Para elementos con características comunes como las animaciones de los personajes, elementos decorativos... serán elementos los cuales puedan ser englobados todos en un solo “Sprite” denominado Spritesheet, con ello podremos ahorrar memoria y nos facilitará la búsqueda de sprites. (Unity nos permite dividir un spritesheet en varios sprites, de forma automática o manual).

Los objetos que interactúan en el juego, son llamados GameObjects (cada uno de los elementos que compondrán nuestro juego), estos pueden ser muy diversos y pueden ir desde nuestro personaje principal hasta el lanzador de un script como por ejemplo, cambiar de nivel (Los componentes de los GameObjects pueden llegar a ser muy diversos). Pero siempre, cada uno de ellos, tendrá la característica <<transform>> en común, esta muestra la posición/rotación/escala del GameObject en cuestión.

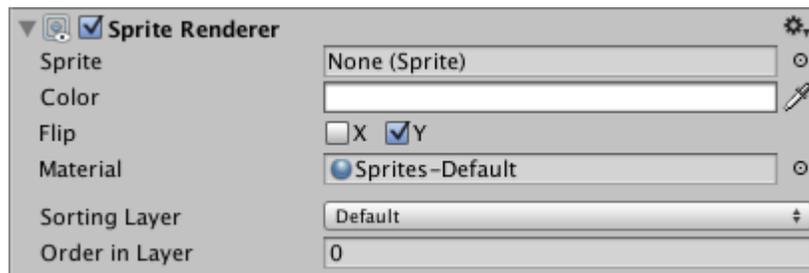


Desde el inspector, a estos GameObjects se les pueden añadir componentes para ampliar sus posibilidades, estos componentes pueden ser muy diversos (Scripts, UI, render, físicas, audio...) y, Unity facilita el uso de estos mediante la creación de “Prefabs” o plantillas cuya función es la de que una vez se modifique un GameObject tendremos la posibilidad de guardar su estado actual (con todos sus componentes y modificaciones) para su posterior o posteriores usos.

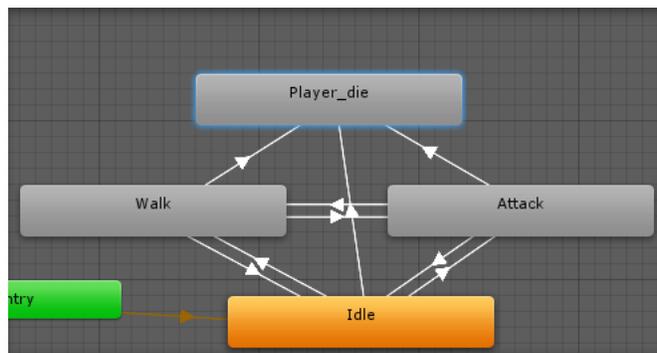
Cabe destacar la posibilidad de modificación de estos prefabs, pero, con cuidado de no aplicar después los cambios en el prefab puesto que esto nos actualizaría todos los utilizados.

Algunos de los componentes más representativos a conocer podrían ser:

- **Sprite Renderer:** Convierte un objeto en un sprite2D



- **Animator:** Nos permitirá crear a partir de distintas animaciones (**Animation**) distintas acciones dentro de nuestro juego correspondiendo con dichas animaciones.



- **Physics 2D:** Permite que dos GameObject colisionen, y el area dependerá tanto del tamaño como del collider utilizado (Box/Polygon...).
- **Rigidbody2D:** Permite a un objeto interactuar con las físicas el juego, ya sea por la fuerza de la gravedad, o fuerzas empleadas para mover a nuestro personaje entre otras.
- **Camera:** Objeto cámara, útil para la visualización del juego, ya sean pantallas de pausa/menú o de juego.

- **Script:** Archivo en lenguaje C#(en este caso) el cual dota a un objeto con distintas funcionalidades y comportamientos.

Cabe destacar que, normalmente, al cambiar entre escenas, todos los objetos de está son destruidos, a no ser que mediante el uso de scripts se especifique que no se quiere destruir determinado GameObject.

Estos, scripts, en todo momento deben heredar de la clase MonoBehaviour, clase principal en Unity, la cual les proporciona distintas funciones que siguen un ciclo de vida en el que las fases basicas podrían ser:

- 1. Awake()** – Exactamente igual que Start(), inicializa componentes de objetos, este solo se llama una vez durante la vida de la instancia del script(independientemente que esté este habilitado o no) y, se llama antes que Start(). Además, es llamado después de que todos los objetos sean inicializados para una cominucación efectiva entre objetos.
- 2. OnEnable()** – Es llamado cuando el objeto se habilita/activa.
- 3. Start()** – Se ejecuta una sola vez al habilitar un script (una vez en la vida del script) y antes que cualquiera de los métodos Update se llame por primera vez.
- 4. Update()** – se ejecuta una vez cada frame. Es la función más utilizada para implementar cualquier tipo de comportamiento del juego.
- 5. FixedUpdate()** –se puede ejecutar más de una vez por cada frame,se utiliza en lugar de Update() cuando se trata de manejar el componentes relacionados con la física, como podría ser - Rigidbody -.
- 6. OnGUI()** –se puede ejecutar varias veces por frame y se llama para procesar y manejar eventos GUI.
- 7. LateUpdate()** –es llamado en cada frame y después de que sean llamadas todas las funciones de actualización (Update)
Ej: Camara: realiza seguimiento de los objetos que podrían haberse movide dentro de un Update.
- 8. OnDisable()** –es llamado cuando el objeto se deshabilita/desactiva.
- 9. OnDestroy()** –es llamado una vez durante la vida del script, solo podrá se llamado en objetos que hayan sido previamente activados y esté sera destruido.

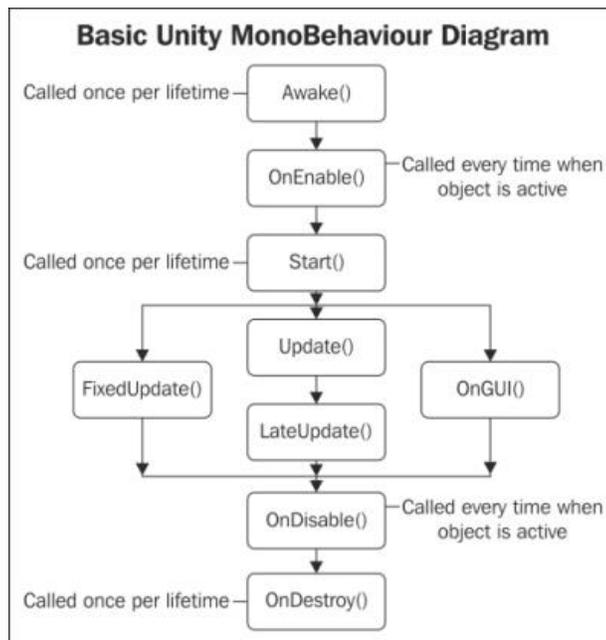


Figura 4. Ciclo de vida de Monobeavior

Cabe destacar que, normalmente, al cambiar entre escenas, todos los objetos de está son destruidos, a no ser que mediante el uso de scripts se especifique que no se quiere destruir determinado GameObject.

La arquitectura de Unity3D (Figura 6) se basa en componentes, permitiendo que esta ser modular y extensible. Esto quiere decir que la comunicación que realiza Unity con la lógica del juego (Controlada por mono) se realiza a través de los componentes. El flujo de ejecución de los scripts es controlado por el motor gráfico, el cual es controlado por el Runtime de Mono. Unity3D compila los assets y sus componentes a través del SDK de Android.



Figura 5. Arquitectura de Unity3D

A partir de la versión 4.6 de Unity encontramos un nuevo sistema de Interfaz de Usuario que permite, de forma rápida y sencilla crear los menús y componentes de la IU, adaptándose a los distintos dispositivos en los que puede ser ejecutado, esto ahorra mucho trabajo debido a que no hará falta la creación de ningún script que adapte el juego a las distintas resoluciones, sino que se hará de forma automática por Unity.

Capítulo 4

Planificación y presupuesto

En este capítulo se describe la metodología elegida para el desarrollo del proyecto, la planificación y el presupuesto del proyecto.

4.1 Metodología

En esta sección se describirán la metodología seguida en el desarrollo del proyecto.

Como metodología se usará, **Waterfall Process**, proceso de desarrollo de software especializado, se trata de una secuencia de diseño de procesos en el cual el desarrollo se basa en el modelo Cascada, consiguiendo tener todo organizado a-priori, lo cual ayudará en la ejecución de todas las etapas siguientes del desarrollo del videojuego (se caracteriza por estar enfocado en el orden y desarrollo riguroso de cada una de sus fases)

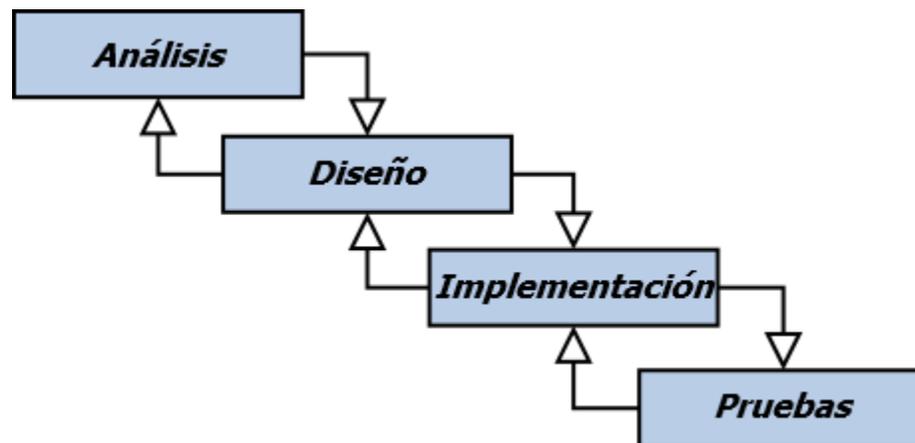


Figura 6. Metodología Waterfall Process

4.2 Planificación

En esta sección se describirán brevemente las fases del desarrollo del proyecto.

Durante el desarrollo del proyecto, va a haber varias fases:

- **Planificación:** La primera que nos podemos encontrar, debido a que a partir de ellas se organizarán las demás.
- **Análisis:** Analizar todos las posibles opciones sobre el tema, la historia, personajes, escenarios, requisitos a tener en cuenta (cualquier tipo de ellos), herramientas y demás variables que puedan afectar al desarrollo del videojuego. Esta fase será realizada por el analista del proyecto.
- **Arte:** Diseño del videojuego de acuerdo al análisis previo. Esta fase sera realizada por el diseñador del proyecto.
- **Desarrollo:** Implementación de los diseños del videojuego con posibles funcionalidades nuevas. Esta fase será realizada por el programador del proyecto.
- **Pruebas:** Pruebas de funcionamiento y rendimiento. Esta fase será realizada por el programador del proyecto.
- **Documentación:** Se elaborara la documentación, memoria, manuales... es decir, toda la información acerca de como se ha realizado, con qué se ha realizado y toda la información relevante con el objetivo de comprender mejor el TFG. Esta fase sera realizada por el documentador del proyecto.
- Cabe destacar que habría que adicionar dos fases más pero debido a que se salen del alcance del proyecto no se llevaron a cabo.
 - **Implementacion:** Compilación del juego y puesta en marcha por parte de los usuarios.
 - **Mantenimiento y soporte:** Seguimientos al videojuego para generar paquetes de correcciones y extensiones de mejora,

Ahora, después de los explicados anteriormente se va a proceder a mostrar y describir el **Diagrama de Gantt (Diagrama 1)**.

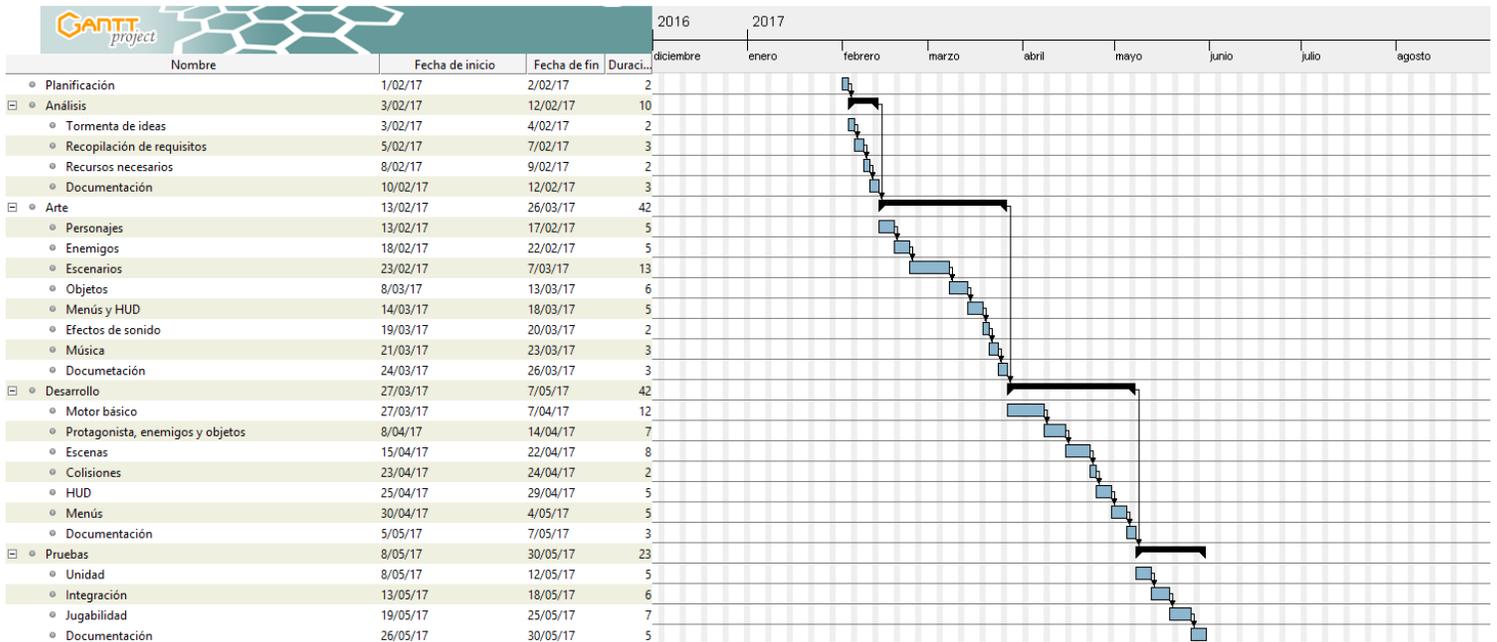


Diagrama 1. Diagrama de Gantt

- **Planificación:** Primera etapa en la que se tratara de definir las ideas principales del proyecto así como las herramientas a utilizar.
 - **Análisis:**
 - **Tormenta de ideas (“Brain Strom”):** Definir el estilo del juego, estética e historia, así como el boceto del personaje principal.
 - **Recopilación de requisitos:** Analizar las necesidades de los posibles usuarios para poder determinar así qué necesidades deberá cubrir el juego,
 - **Recursos necesarios:** Analizar qué recursos serán necesarios para cubrir con satisfacción dichas necesidades.
 - **Documentacion:** Documentar lo realizado hasta el momento.
 - **Arte:**
 - **Personajes:** Diseño del personaje principal, así como sus animaciones y demás elementos orientados a él.
 - **Enemigos:** Diseño de los enemigos del personaje principal, así como su animaciones y demás elementos orientados a ellos.
 - **Escenarios:** Diseño de los distintos tipos de escenarios.
 - **Objetos:** Diseño de todo lo relativo a objetos/items decorativos, ya sea con el objetivo de dedicarles a objetos de misión, u objetos de personaje o simplemente objetos decorativos a añadir en los distintos Escenarios.

- **Menús y HUD:** Diseño de los distintos tipos de menus que nos podremos encontrar a lo largo del juego y, además, del HUD (información en pantalla relativa a nuestro personaje)
- **Efectos de Sonido:** Diseño de los efectos de sonidos del juego.
- **Música:** Diseño de la música ambiental del juego.
- **Documentación:** Documentar lo realizado hasta el momento.

● **Desarrollo:**

- **Motor básico:** Familiarización con el motor Unity3D y sus componentes.
- **Protagonista/enemigos/objetos:** Importar a Unity3D el personaje/enemigos y objetos diseñados e implementar las funcionalidades básicas del videojuego.
- **Escenas:** Importar todos los objetos relacionados con las escenas.
- **Colisiones:** Probar las animaciones de la GUI del juego y su correcta respuesta.
- **HUD:** Implementar los HUD de cada modo de juego.
- **Menús:** Implementar todos los menus disponibles (menu principal, configuración, pausa...)
- **Documentación:** Documentar lo realizado hasta el momento.

● **Pruebas:**

- **Unidad:** Probar el funcionamiento individual de cada elemento para la posterior integración.
- **Integración:** Consiste en comprobar que todos los objetos en su conjunto se comportan de la manera esperada.
- **Jugabilidad:** Conjunto de propiedades que describen la experiencia del jugador ante un sistema de juego determinado, cuyo principal objetivo es divertir y entretener.
- **Documentación:** Documentar lo realizado hasta el momento.

4.3 Presupuesto inicial

En esta sección se describirá la estimación del presupuesto del proyecto.

Siguiendo la planificación definida en el Diagrama de Gantt (Diagrama1), se estima el presupuesto del proyecto.

Se va a proceder a organizar dicho presupuesto mediante la división de los distintos posibles presupuestos:

- **Presupuesto hardware:** Elementos hardware que serán necesarios durante el desarrollo del proyecto.

- **Presupuesto software:** Software necesario durante el desarrollo del proyecto.
- **Presupuesto de desarrollo:** Recursos humanos necesarios durante el desarrollo del proyecto.

4.3.1 Presupuesto hardware

Esta sección tratara sobre los componentes hardware.

Hardware necesario para el desarrollo del proyecto:

- **Ordenador personal:** Para la realización del proyecto, unos de los elementos principales a tener en cuenta sera el contar con un ordenador, en este caso, el sistema operativo con el que se contará será Windows.

Para calcular su uso, debera tenerse en cuenta que la media de vida de un ordenador, en la actualidad, debido a los grandes cambios que ocurren en tan poco tiempo en lo referido a esta tecnología es aproximadamente de 4 años y ya que se ha hecho uso de él para este proyecto durante 16 semanas – 4meses se calcularía:

$$\begin{array}{r}
 100\% \text{ ----- } 192 \text{ semanas} \\
 x\% \text{ ----- } 16 \text{ semanas} \\
 x = (16 \cdot 100) / 192 = \mathbf{8,34\%} \text{ de uso.}
 \end{array}$$

- **Periféricos del ordenador:** Para la realización del proyecto, se ha hecho uso de otros elementos que complementan al ordenador, como el uso de un ratón para manejarlo facilmente, el uso de un disco duro externo el cual sera el encargado de almacenar todas las copias de seguridad yel uso de una impresora.
 - **Impresora HP Deskjet F4500 series class :** Para la documentación se requerirá de una impresora, el gasto de esta sera de 50€ y se estima gastar en tinta 60€, haciendo uso del **100%**.
 - **Ratón:** Para manejar de forma eficiente el ordenador se utilizará un ratón cuyo coste es de 14€ y con un uso del **100%**.
 - **Disco duro externo:** Para almacenar las distintas copias de seguridad que se iran produciendo a lo largo del proyecto utilizaremos un disco duro externo cuyo coste es de 60€ y con un uso del **100%**.

- **Dispositivo móvil:** Para poder ir probando el videojuego, se necesitara hacer uso de un dispositivo movil con el sistema operativo, Android. En el cuál se instalara la aplicación.
Para calcular su uso, utilizamos el mismo método utilizado anteriormente, con el apunte que la vida media de un móvil hoy en día es de 3 años, el periodo en el que se utilizará para el proyecto es el mismo.

$$100\% \text{ ----- } 144 \text{ semanas}$$

$$x\% \text{ ----- } 16 \text{ semanas}$$

$$x = (16 * 100) / 144 = \mathbf{11.12\%}$$
 de uso.
- **Cable microUSB:** Se encargara de permitirnos la conexión movil-ordenador para llevar a cabo el traspaso de la aplicación al dispositiivo.
Este ha sido elegido debido a ser compatible con el tipo de dispositivo movil, su bajo coste y, su porcentaje de uso será el 50%.
- **Conexión a Internet:** Para la realización del proyecto se necesitara de conexión a internet, esta dependera del precio de la compañía, suponiendo que se trata de “ONO”, el precio de la banda ancha es de 49€, siendo pagado por mes, se considera un 40% de su uso mensual.
- **Material de oficina:** Para la elaboración de ideas generales, notas, diagramas...

Componente Hardware	Uso (%)	Coste Total(%)	Coste Desarrollo (Uso*Coste)/100
Ordenador Portatil			
Toshiba Satellite C55-A-1H9	8.34%	480	40.032
Dispositivo Móvil			
BQ E4.5	11.12%	160	17.792
Perifericos			
Ratón(Krom Khanda 4000DPI)	100%	30	30
Cable microUSB	50%	1.7	0.85
Impresora HP Deskject F4500	100%	50	50+60(Copias-tinta)
Material de oficina			
Cuadernos, bolígrafos...	100%	10	10
Conexión a Internet			
Conexión 50MB	50%	49*4	98
TOTAL(€):		306.674€	

Tabla 1. Presupuesto Inicial – Presupuesto hardware

4.3.2 Presupuesto software

Esta sección tratara sobre los componentes software.

Software necesario para el desarrollo del proyecto:

- **Unity3D, Android SDK, Monodevelop, Dia, StarUML, Gimp, Draw.io, FL Studio, UnityRemote** son gratuitos, por lo que el coste es de 0€.
- **Microsoft Office 2007 Professional Plus:** el coste de la licencia de este paquete es de 52€ y se usa el mismo porcentaje que el ordenador donde se instala, **8.34%**.

Componente Hardware	Uso (%)	Coste Total(%)	Coste Desarrollo (Uso*Coste)/100
Unity3D	-	-	0
Android SDK	-	-	0
Monodevelop	-	-	0
Unity Remote	-	-	0
GIMP	-	-	0
Dia	-	-	0
Draw.io	-	-	0
FL Studio	-	-	0
StarUML	-	-	0
Microsoft Office Professional Plus 2007	8.34%	52	4.34
TOTAL(€):		4.34€	

Tabla 2. Presupuesto inicial – Presupuesto software

4.3.3 Presupuesto de desarrollo

Esta sección tratara sobre los costes del desarrollo del proyecto por parte de un Ingeniero Informático de Servicios y Aplicaciones.

Debido a que en este caso, en el desarrollo de este proyecto, el autor es el único implicado, esté ejercerá los distintos roles:

- **Analista:** Parte encargada de la fase de Análisis del proyecto
- **Diseñador:** Parte encargada de la fase de Diseño del proyecto

- Programador: Parte encargada de la fase de Implementación y Pruebas del proyecto.
- Documentador: Parte encargada de la fase de Documentación del proyecto.

La duración del proyecto ha sido de aproximadamente de 5 meses trabajando 8 horas al día, y los costes de desarrollo.

Contando que los meses tienen 22 días laborables:

$$117 \text{ días} * 8\text{h/día} = \mathbf{936\text{h}}.$$

Contando que se deberán realizar todos los roles por el autor del proyecto, se podrá calcular el presupuesto mirando en el diagrama de GANTT los días/horas que cada rol está presente en el desarrollo del proyecto,

Rol	Tiempo	Coste
Analista	56h	10€/h
Diseñador	312h	10€/h
Programador	456h	12€/h
Documentador	112h	8€/h
TOTAL(€):	10048€	

Tabla 3. Presupuesto inicial- Presupuesto de desarrollo

4.3.3 Presupuesto total

El presupuesto total se obtendrá mediante la suma total de los presupuestos calculados anteriormente:

Presupuesto	Coste(€)
Hardware	306.674
Software	4.34
Desarrollo	10048
TOTAL(€):	10359,014

Tabla 4. Presupuesto inicial – Presupuesto Total

4.4 Estimación de costes mediante CO-COMO

Esta sección tratará sobre la estimación de costes mediante el uso del modelo COCOMO.

Para realizar una estimación de los costes software se utilizará el Modelo Constructivo de Costes (Cocomo).

Este se trata de un modelo matemático desarrollado por Barry W. Boehm y está orientado al producto final, midiendo la cantidad de líneas de código del proyecto y una serie de factores de complejidad.

Se pueden presentar tres subniveles: básico, intermedio y detallado

- **Básico:** Este modelo trata de estimar de una manera rápida y más o menos burda, la mayoría de proyectos pequeños y medianos. Se consideran tres modos de desarrollo de este modelo: orgánico empotrado, semiempotrado
- **Intermedio:** En este modelo se introducen 15 atributos de coste para tener en cuenta el entorno de trabajo. Estos atributos se utilizan para ajustar el coste nominal del proyecto al entorno real, incrementando la precisión de la estimación.
- **Detallado:** Este modelo puede procesar todas las características del proyecto para construir una estimación.

Además, se pueden distinguir tres modelos diferentes:

- **Orgánico:** Proyectos software desarrollados por un pequeño grupo de programadores en un entorno familiar. El tamaño del software varía pero no sobrepasan las decenas de miles de líneas de código.
- **Empotrado:** Se trata de proyectos software que cuentan con fuertes restricciones en lo relativo al procesamiento o la interfaz. El problema a resolver es único y es difícil basarse en la experiencia, puesto que al ser único se carece de ella.
- **Semiempotrado:** Corresponde a un esquema intermedio entre los modelos: orgánico y empotrado. Dependiendo del problema, el grupo puede incluir una mezcla de personas experimentadas y no experimentadas.

Para este proyecto se elegirá un submodelo intermedio debido a que se dispone de una especificación de requisitos, este submodelo requerirá aplicar 15 factores de coste.

Las líneas de código que se estiman para este proyecto serán “”4,5 KLDC (miles de líneas de código) y, debido a esto, se escogerá el modelo Orgánico, ya que el tamaño del proyecto, como se ha comentado anteriormente, se ajustan a ello.

La siguiente tabla (Tabla 5), contiene los 15 factores de coste, necesarios para la obtención de los factores de complejidad:

FACTORES DE COSTE	VALOR					
	Muy bajo	Bajo	Medio	Alto	Muy alto	Extra
Factores software						
Fiabilidad requerida	0,75	0,88	1,00	1,15	1,4	-
Tamaño de la Base de Datos	-	0,94	1,00	1,08	1,16	-
Complejidad del software	0,70	0,85	1,00	1,15	1,30	1,65
Factores hardware						
Restricciones de rendimiento en tiempo de ejecución	-	-	1,00	1,11	1,30	1,66
Restricciones de memoria	-	-	1,00	1,06	1,21	1,56
Volatilidad del entorno de la máquina virtual	-	0,87	1,00	1,15	1,30	-
Tiempo de respuesta	-	0,87	1,00	1,07	-	-
Factores de personal						
Capacidad de los analistas	1,46	1,19	1,00	0,86	0,71	-
Experiencia con el tipo de aplicación	1,29	1,13	1,00	0,91	0,82	-
Experiencia con el hardware	1,21	1,10	1,00	0,90	-	-
Experiencia con el lenguaje de programación	1,14	1,07	1,00	0,95	-	-
Capacidad de los programadores	1,42	1,17	1,00	0,86	0,70	-
Factores del proyecto						
Técnicas modernas de programación	1,24	1,10	1,00	0,91	0,82	-
Utilización de herramientas software	1,24	1,10	1,00	0,91	0,83	-
Restricciones en la planificación temporal del desarrollo	1,23	1,08	1,00	1,04	1,10	-

Tabla 5. Factores de coste COCOMO

Ahora se calcula el **factor de ajuste del esfuerzo (m(X))** que es el productorio de los 15 factores definidos anteriormente:

$$m(\mathbf{X}) = \prod_{k=1}^n \text{factor } k = 1,15 * 1,15 * 1,30 * 1,06 * 1,00 * 1,07 * 1,00 * 1,00 * 0,90 * 1,00 * 1,00 * 0,91 * 0,91 * 1,04 = 1.51142$$

Se siguen los siguientes factores de complejidad que encontramos en la siguiente tabla debido a que el modelo es de tipo Orgánico.

	a	b	c	d
Versión	3,2	1,05	2,5	0,38
Descripción	3,0	1,12	2,5	0,35
Dependencias	2,8	1,20	2,5	0,32

Tabla 6. Factores de complejidad COCOMO

Estos factores se utilizan para calcular el **esfuerzo (E)**:

$$E = a * KLDC^b * m(X) = 3,2 * 4,5^{1,05} * 1,51142 = 23,46 \frac{\text{personas}}{\text{mes}}$$

El **tiempo de desarrollo (TD)** equivale a:

$$TD = c * E^d = 2,5 * 23,46^{0,38} = 8,292 \text{ meses.}$$

Finalmente se calcula el **número medio de personas** necesarias para realizar el proyecto:

$$\text{Nº medio de personas} = \frac{E}{TD} = \frac{23,46 \frac{\text{personas}}{\text{mes}}}{8,292 \text{ meses}} \approx 2,829 \text{ personas.}$$

Tras realizar los anteriores calculos mediante COCOMO se puede estimar que el proyecto deberá ser desarrollado por “3” personas trabajando durante 8meses.

Por contra, el proyecto será desarrollado solo por una persona, o si que el tiempo se calcularía: $8*3=24$ meses.

Para poder estimar los costes del proyecto mediante este modelo, se necesita conocer el total de horas a trabajar:

$$(24\text{meses}*4\text{semanas}*5\text{días})*8\text{h} = 480\text{días}*8\text{h} = \mathbf{3840\text{h}}$$

Se estima el esfuerzo de las distintas categorías de trabajo en la siguiente tabla, cabe destacar que son estimaciones y puede variar en decimas del resultado.

Rol	Esfuerzo(%)
Analista	6%
Diseñador	33%
Programador	49%
Documentador	12%
TOTAL(€):	100%

Tabla 7. Esfuerzo categorías de trabajo

4.4.1 Estimación costes hardware

Esta sección tratará sobre los componentes hardware, se utilizan los datos de la vida media usados en el Presupuesto Inicial, pero, esta vez, para los datos obtenidos mediante el CO-COMO.

Hardware necesario para el desarrollo del proyecto:

- **Ordenador personal:** Según COCOMO se hará uso de él durante 24 meses, que son 96 semanas:

$$\begin{array}{r} 100\% \text{ ----- } 192 \text{ semanas} \\ x\% \text{ ----- } 96 \text{ semanas} \\ x = (96*100)/192 = \mathbf{50\%} \text{ de uso.} \end{array}$$

- **Periféricos del ordenador:** Para la realización del proyecto, se ha hecho uso de otros elementos que complementan al ordenador, como el uso de

un ratón para manejarlo fácilmente, el uso de un disco duro externo el cual sera el encargado de almacenar todas las copias de seguridad y el uso de una impresora.

- **Impresora HP Deskjet F4500 series class** : Para la documentación se requerirá de una impresora, el gasto de esta sera de 50€ y se estima gastar en tinta 60€, haciendo uso del **100%**.
- **Ratón**: Para manejar de forma eficiente el ordenador se utilizará un ratón cuyo coste es de 14€ y con un uso del **100%**.
- **Disco duro externo**: Para almacenar las distintas copias de seguridad que se irán produciendo a lo largo del proyecto utilizaremos un disco duro externo cuyo coste es de 60€ y con un uso del 100%.

- **Dispositivo móvil**: Igual que el ordenador se hará uso de el durante 96 semanas:

$$\begin{array}{r} 100\% \text{ ----- } 144 \text{ semanas} \\ x\% \text{ ----- } 96 \text{ semanas} \\ x = (96*100)/144 = \mathbf{66.6\%} \text{ de uso.} \end{array}$$

- **Cable microUSB**: Se hará uso de el al 100%.
- **Conexión a Internet**: Debido a que, como se ha descrito antes, se considera un gasto del 50% por mes, habría que calcularlo para los 24.
- **Material de oficina**: Para la elaboración de ideas generales, notas, diagramas...

Componente Hardware	Uso (%)	Coste Total(%)	Coste Desarrollo (Uso*Coste)/100
Ordenador Portatil			
Toshiba Satellite C55-A-1H9	50%	480	240
Dispositivo Móvil			
BQ E4.5	66.6%	160	106.56
Perifericos			
Ratón(Krom Khanda 4000DPI)	100%	30	30
Cable microUSB	100%	1.7	1.7
Impresora HP Deskject F4500	100%	50	50+60(Copias-tinta)
Material de oficina			
Cuadernos, bolígrafos...	100%	10	10
Conexión a Internet			
Conexión 50MB	50%	49*24	1176
TOTAL(€):		1674,26€	

Tabla 8. Estimación de costes hardware –COCOMO-

4.4.2 Estimación costes software

Esta sección trata sobre los componentes software junto con los datos obtenidos del COCOMO

Software necesario para el desarrollo del proyecto:

- **Unity3D, Android SDK, Monodevelop, Dia, StarUML, Gimp, Draw.io, FL Studio, UnityRemote** son gratuitos, por lo que el coste es de 0€.
- **Microsoft Office 2007 Professional Plus:** el coste de la licencia de este paquete es de 52€ y se usa el mismo porcentaje que el ordenador donde se instala, **50%**.

Componente Hardware	Uso (%)	Coste Total(%)	Coste Desarrollo (Uso*Coste)/100
Unity3D	-	-	0
Android SDK	-	-	0
Monodevelop	-	-	0
Unity Remote	-	-	0
GIMP	-	-	0
Dia	-	-	0
Draw.io	-	-	0
FL Studio	-	-	0
StarUML	-	-	0
Microsoft Office Professional Plus 2007	50%	52	26
TOTAL(€):		26€	

Tabla 9. Estimación de costes software –COCOMO-

4.4.3 Presupuesto de desarrollo

Esta sección tratara sobre los costes del desarrollo del proyecto por parte de un Ingeniero

La duración del proyecto según el COCOMO sido de 24meses trabajando 8horas al día, y los costes de desarrollo.

Contando que los meses tienen 22 días laborables:

$$528 \text{ días} * 8\text{h/día} = 4224\text{h.}$$

Para calcular el presupuesto en función de los roles se hará uso de la tabla7 que hace referencia al esfuerzo de cada rol, así, obtenemos las horas estimadas de cada rol:

Rol	Tiempo	Coste
Analista	6%*4224h	10€/h
Diseñador	33%*4224h	10€/h
Programador	49%*4224h	12€/h
Documentador	12%*4224h	8€/h
TOTAL(€):		45365,76€

Tabla 10. Presupuesto inicial – Presupuesto de desarrollo

4.4.4 Presupuesto Total

El presupuesto total se obtendrá mediante la suma total de los presupuestos calculados anteriormente:

Presupuesto	Coste(€)
Hardware	1674,26
Software	26
Desarrollo	45365,76
TOTAL(€):	47066.02

Tabla 11. Presupuesto inicial – Presupuesto total

4.5 Conclusiones

El presupuesto inicial se planteo para un proyecto con una duración de “5” trabajando 8 horas al día, durante los 22 días laborables que tiene un mes. Este primer cálculo de tiempo (5meses) no se ajusta del todo, debido a que se atrasó unos días por la magnitud del proyecto, esto, siempre hace que el proyecto se encarezca un poco más (hay que contar que se ampliaría el uso de elementos tanto software como software, además, del sueldo)

La diferencia abismal entre la estimación del COCOMO se debe principalmente a que la estimación mediante este método nos da como resultado la necesidad de realización del proyecto por parte de 3 personas, trabajando durante 8 meses. Esto esta “cercano” a lo que ha llevado en la realidad, dista poco, pero debido a que se trata solo de una persona realizándolo, se dispara a los 24 meses, plazo imposible de seguir, aumentando como se ha expuestó antes, tanto el sueldo como la utilización de componentes hardware y software.

TIPO	Coste(€)
Presupuesto Inicial	10359,014
Estimación de costes mediante COCOMO:	47066.02

Tabla 12. Comparación resultados

Capítulo 5

Análisis

En este capítulo se realiza la especificación del comportamiento y funcionalidades del sistema, mediante el uso del modelado UML

Antes de comenzar, se expondrá una “leyenda” para poder comprender con más facilidad las tablas:

- **ID:** Identificador principal.
- **Nombre:** Identifica el contenido.
- **Versión:** Número de versión.
- **Actor:** Usuario que utiliza el contenido.
- **Disparador:** Evento que inicia el contenido.
- **Dependencias:** Relaciones con otras tablas.
- **Precondiciones:** Condiciones previas que deben cumplirse para poder utilizarse.
- **Prioridad:** Indica la importancia, se especifica como Baja/Media/Alta/Muy Alta.
- **Estado:** Estado en el que se encuentra el contenido, se especifica como Implementado/No implementado, en el caso de estar implementado solo para ciertas funcionalidades se usarán ambas juntas [Implementado/no implementado]
- **Frecuencia de uso:** Indica la utilización, se especifica como Baja/Media/Alta/Muy Alta.
- **Flujo normal:** Secuencia de pasos que sigue el usuario al utilizar el contenido, se especifica como Paso (número)/Acción.
- **Flujo alternativo:** Secuencia alternativa de pasos que sigue el usuario al utilizar el contenido, se especifica como Paso (número)/Acción.
- **Postcondiciones:** Condiciones que modifica el uso del contenido.
- **Excepciones:** Excepciones que pueden aparecer durante el uso del contenido.
- **Descripción:** Identifica el contenido de forma detallada.
- **Otra información:** Otra información necesaria.

5.1 Identificación de actores

En esta sección se identifica a los actores a los que irá dirigido el producto.

Todos los usuarios serán jugadores, por lo que existe un único actor.

Id y Nombre	ACT-01 Jugador
Versión	V1
Descripción	Usuario/Jugador de la aplicación.
Otra información	Comentarios

Tabla 13 ACT-01 Jugador

5.2 Objetivos

En esta sección se identifican los objetivos del proyecto.

La siguiente tabla, muestra una plantilla de cómo se definirán los objetivos:

Id y Nombre	OBJ-XY
Versión	Número de Versión
Descripción	Descripción detallada del contenido
Prioridad	Nivel de importancia en la implementación
Estado	Implementado/No implementado
Otra información	Comentarios

Tabla 14. .Plantilla Ejemplo de tabla Objetivos

Objetivos:

Id y Nombre	OBJ-01 Diseño del Videojuego
Versión	v1
Descripción	Implementación de la estructura general del videojuego, permitiendo crear un menú “multi-juego” y las distintas características sobre ellos.
Prioridad	Muy Alta
Estado	Implementado
Otra información	Comentarios

Tabla 15. OBJ-01 Diseño del videojuego

Id y Nombre	OBJ-02 Creación de la Interfaz de Usuario
Versión	v1
Descripción	Se implementarán los menús necesarios para facilitar la interacción “juego-usuario”.
Prioridad	Muy alta
Estado	Implementado
Otra información	Estos menús son: - Menú Principal (Elección de juego + botón configuración + botón salida) - Menú configuración (Gestión de música) - Menú de pausa → Modo desafío - Menú GameOver → Modo plataforma

Tabla 16. OBJ-02 Creación de la GUI

Id y Nombre	OBJ-03 Programación de las funcionalidades del videojuego
Versión	v1
Descripción	Se implementaran las funcionalidades básicas del videojuego
Prioridad	Muy Alta
Estado	Implementado
Otra información	Funcionalidades: - Mini-juego al estilo plataforma: - Movimiento del personaje por el nivel “infinito” - Actualizar el HUD con la puntuación que se va obteniendo en tiempo real, la puntuación se obtendrá por bloques “pequeños” avanzados(+1 punto) y por cristal recogido(+3 puntos) - Generador y destructor de bloques para la optimización de memoria. - Modo desafío - Movimiento del personaje - Actualización HUD vida personaje - Inicio/fin de misiones - Modo Historia: - No implementado

Tabla 17. OBJ-03 Programación de las funcionalidades del videojuego

Id y Nombre	OBJ-04 Implementación Multi-Juego
Versión	v1
Descripción	Se implementara un menú principal en el que se podrá seleccionar a cuál de los 3 estilos de juego juga.
Prioridad	Alta
Estado	Implementado (modo campaña no)
Otra información	<p>El juego va a estar compuesto internamente por 3 sub-juegos.</p> <ul style="list-style-type: none"> -Mini-Juego al estilo plataformas, en el que el personaje y su montura deben recoger el mayor número de cristales posible, no hay límite de tiempo o de espacio por recorrer y la puntuación final dependerá del número de plataformas recorridas y del número de cristales recogidos. -Modo-Desafío: El género de este modo se podría denominar << Action-RPG>>. Se caracterizará por tener que superar una serie de niveles, alternando entre niveles de desafío intelectual y niveles “Horda” en los que el jugador deberá aguantar hasta determinada ronda evitando morir por los enemigos, los cuales reaparecer en cada ronda. -Modo-Historia: El género de este modo se trata de un juego <<RPG>>. Se caracterizará por tener un mapa de mundo muy extenso y una historia que guíe a nuestro personaje hasta el final, si le hubiera. <p>-Los tres modos de juego se complementarán debido a que solo se podrá completar el desarrollo del modo historia al 100% pasándose todos los modos de juego. (Cabe destacar que el mini-juego no tendrá fin, pero en un futuro se podrá desbloquear ciertas cosas al llegar a una determinada puntuación. Ejemplo: monturas)</p>

Tabla 18. OBJ-04 Implementación Multi-Juego

Id y Nombre	OBJ-05 Creación Nivel “Infinito”
Versión	v1
Descripción	Implementación de un nivel infinito, que se autogenera para que nuestro personaje pueda avanzar, pero también se autodestruya para evitar una carga excesiva en memoria
Prioridad	Alta
Estado	Implementado
Otra información	Modo: Mini-Juego(Plataforma)

Tabla 19. OBJ-05 Creación Nivel “Infinito”

Id y Nombre	OBJ-06 Creación de distintos niveles
Versión	v1
Descripción	Descripción detallada del contenido
Prioridad	Media
Estado	No implementado
Otra información	<p>Modo: Desafío (Action-RPG)</p> <p>-Sólo se ha implementado para esta versión 1 nivel</p>

Tabla 20. OBJ-06 Creación de distintos niveles

Id y Nombre	OBJ-07 Creación Modo Historia
Versión	V1
Descripción	Implementación de un modo historia en el que, nuestro personaje tendrá que recorrer un largo camino descubriendo los entresijos de la historia.
Prioridad	Media
Estado	No implementado
Otra información	Comentarios

Tabla 21. OBJ-07 Creación Modo Historia

Id y Nombre	OBJ-08 Cargar progreso
Versión	v1
Descripción	Implementación de la función de cargado.
Prioridad	Media
Estado	Implementado/No implementado
Otra información	Actualmente, la única función de cargado disponibles es la que encontramos en el modo plataforma, se encargara de cargar automáticamente la máxima puntuación obtenida en anteriores partidas para su superaci

Tabla 22. OBJ-08 Cargar progreso

Id y Nombre	OBJ-09 Guardar Progreso
Versión	v1
Descripción	Implementación de la función guardado
Prioridad	Media
Estado	Implementado/No implementado
Otra información	Actualmente, la única función de guardado disponible es la que encontramos en el modo plataforma, el cual guardara el máximo número de puntos antes de la muerte del personaje, incitando a cada usuario a superar su record actual.

Tabla 23. OBJ-09 Guardar Progreso

Id y Nombre	OBJ-10 Creación del sonido el juego
Versión	v1
Descripción	Se implementara la música y efectos de sonido del videojuego
Prioridad	Media
Estado	Implementado
Otra información	La música será generada mediante la transformación de canciones antiguas de videojuegos con FL Studio. Los efectos de sonido serán obtenidos desde una fuente externa.

Tabla 24. OBJ-10 Creación del sonido de juego

Id y Nombre	OBJ-11 Video “Intro” del juego
Versión	v1
Descripción	Implementación de un video introductorio del juego, este video será generado sin la utilización de fuentes externas.
Prioridad	Media
Estado	Implementado
Otra información	Posible adición de sonido en un futuro

Tabla 25. OBJ-11 Video “Intro” del juego

5.3 Especificación de Requisitos

En esta sección se describen los requisitos de la aplicación, dando a conocer las funcionalidades a desarrollar y la información necesaria para el conocimiento del videojuego.

La siguiente tabla muestra una platilla de cómo van a ir organizados/definidos los requisitos del sistema.

Id y Nombre	RQ[S]-XY
Versión	Número de Versión
Descripción	Descripción detallada del contenido
Dependencias	Requisitos necesarios para su funcionamiento
Prioridad	Nivel de importancia en la implementación
Frecuencia de uso	Uso del contenido
Otra información	Comentarios

Tabla 26. Ejemplo de tabla de Requisitos

5.3.1 Requisitos Funcionales

Nos mostrarán divididos en tablas las funcionalidades que debe tener la aplicación, éstas, a su vez, para facilitar su legibilidad estarán clasificadas dependiendo del modo de juego al que hagan referencia, debido que algunas funcionalidades son exclusivas de cada juego.

Se recogerán aquellos requisitos comunes entre los distintos modos de juego:

Id y Nombre	RQF-01 Seguir al jugador con la Cámara
Versión	v1
Descripción	El sistema deberá poder seguir el movimiento del personaje a través de la cámara principal.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 27. RQF-01 Seguir al jugador con la Cámara

Id y Nombre	RQF-02 Colisionar con Objetos
Versión	v1
Descripción	El sistema deberá poder detectar cuando se produzca colisiones entre objetos específicos, y de ser así, generar eventos con funciones específicas.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	Estos objetos, en este caso, se trataran del personaje, las plataformas, ítem(cristal) a recoger por el personaje...

Tabla 28. RQF-02 Colisionar con Objetos

Id y Nombre	RQF-03 Gestor Animaciones
Versión	v1
Descripción	El sistema deberá poder decidir que animación utilizar dependiendo el caso en el que se encuentre nuestro personaje o, en el caso que hubiera enemigos, nuestros enemigos.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	...

Tabla 29. RQF-03 Gestor Animaciones

Id y Nombre	RQF-04 Muerte Personaje
Versión	v1
Descripción	El sistema deberá “matar/destruir” nuestro personaje
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	Mini-juego plataformas: Cuando esté se caiga de las plataformas al vacío. Modo Desafío: Cuando nuestro personaje reciba más daño del que se le permita recibir.

Tabla 30. RQF-04 Muerte Personaje

Id y Nombre	RQF-05 Recoger Item/Cristal
Versión	v1
Descripción	El sistema deberá poder detectar que el personaje recogió un Item.
Dependencias	RQF-02
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	En modo de juego Plataforma: Tras recoger el ítem el sistema deberá actualizar el HUD. Dicha actualización del HUD seguirá el patrón: Número máximo de puntos+ 3.

Tabla 31. RQF-05 Recoger Item/Cristal

Id y Nombre	RQF-06 Reproducir música de fondo
Versión	v1
Descripción	El sistema deberá seguir reproduciendo la canción generada en el menú principal.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	Cabe destacar que en un futuro se espera tener varias canciones a lo largo del juego, no solo una.

Tabla 32. RQF-06 Reproducir música de fondo

Id y Nombre	RQF-07 Reproducir efecto del sonido
Versión	v1
Descripción	El sistema deberá, ante determinadas acciones, reproducir el efecto de sonido designado para ello.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 33. RQF-07 Reproducir efecto de sonido

Id y Nombre	RQF-08 Gestionar música de fondo
Versión	v1
Descripción	El sistema deberá poder activar/desactivar la música del juego.
Dependencias	RQF-06
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	Cabe destacar que en un futuro se espera tener varias canciones a lo largo del juego, no solo una.

Tabla 34. RQF-08 Gestionar música de fondo

Id y Nombre	RQF-09 Menú GameOver
Versión	v1
Descripción	El sistema deberá mostrar al usuario un menú cuando el personaje se “caiga al vacío” o cuando esté, pierda todos sus puntos de vida
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	<p>Este menú contendrá:</p> <ul style="list-style-type: none"> -Modo Plataforma: <ul style="list-style-type: none"> -Panel informativo: Puntuación máxima conseguida, puntuación en partida actual y título GameOver. -Botón reiniciar nivel -Botón para volver al menú de selección de juego. -Modo Desafío: <ul style="list-style-type: none"> -Panel informativo: Título GameOver -Botón reiniciar nivel -Botón para volver al menú de selección de juego

Tabla 35. RQF-09 Menú GameOver

Id y Nombre	RQF-10 Reiniciar Nivel
Versión	v1
Descripción	El sistema deberá poder dar la opción de reiniciar nivel cuando el personaje muera.
Dependencias	
Prioridad	Media
Frecuencia de uso	Alta
Otra información	

Tabla 36. RQF-10 Reiniciar Nivel

Id y Nombre	RQF-11 Salir del juego
Versión	v1
Descripción	El sistema deberá poder dar a opción de salir al menú de selección de juego cuando el personaje muera
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 37. RQF-11 Salir del juego

Mini-juego Estilo Plataforma

Se recogerán las funcionalidades básicas para este modo de juego

Id y Nombre	RQF-12 Mover al personaje
Versión	v1
Descripción	Tras la primera pulsación en pantalla, el sistema deberá mover al jugador a la derecha.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	Comentarios

Tabla 38. RQF-012 Mover al personaje

Id y Nombre	RQF-13 Saltar
Versión	v1
Descripción	El sistema deberá poder aplicar una fuerza sobre nuestro personaje, dotándole de salto.
Dependencias	RQF-12
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	El máximo número de saltos consecutivos realizados por el jugador es de 2.

Tabla 39. RQF-13 Saltar

Id y Nombre	RQF-14 Generar plataformas
Versión	v1
Descripción	Una vez que el personaje se ponga en movimiento, el sistema deberá autogenerar plataformas en 3 niveles para su avance.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	Comentarios

Tabla 40. RQF-14 Generar plataformas

Id y Nombre	RQF-15 Destruir plataformas
Versión	v1
Descripción	Una vez que el personaje se ponga en movimiento, se irán destruyendo las plataformas que esté vaya dejando atrás con el objetivo de reducir la memoria utilizada.
Dependencias	RQF-01, RQF-04
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	Comentarios

Tabla 41. RQF-15 Destruir plataforma

Id y Nombre	RQF-16 Actualizar HUD
Versión	v1
Descripción	El sistema deberá actualizar a tiempo real la puntuación del jugador en el HUD del juego con los Items(Cristales) recogidos y con las plataformas recorridas.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 42. RQF-16 Actualizar HUD

Id y Nombre	RQF-17 Parallax Scrolling
Versión	v1
Descripción	Con el objetivo de intentar conseguir sensación de profundidad, el sistema deberá otorgar distintas velocidades a los distintos componentes de los que está constituido y otorgarles movimiento.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 43. RQF-17 Parallax Scrolling

Id y Nombre	RQF-18 Guardar puntuación máxima
Versión	v1
Descripción	El sistema deberá almacenar la cantidad máxima de puntos obtenidos para así incitar al usuario a mejorar su puntuación
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 44. RQF-18 Guardar puntuación máxima

Id y Nombre	RQF-19 Cargar puntuación máxima
Versión	v1
Descripción	El sistema deberá cargar la cantidad máxima de puntos obtenidos para así mostrar al usuario el record obtenido.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 45. RQF-19 Cargar puntuación máxima

Modo Desafío

Se recogerán las funcionalidades básicas para este modo de juego

Id y Nombre	RQF-20 Mover jugador a la derecha
Versión	v1
Descripción	El sistema deberá poder detectar la posición del joystick, tras esto, si la dirección es derecha, el sistema deberá mover el personaje hacia esa dirección.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 46. RQF-20 Mover jugador a la derecha

Id y Nombre	RQF-21 Mover jugador a la izquierda
Versión	v1
Descripción	El sistema deberá poder detectar la posición del joystick, tras esto, si la dirección es izquierda, el sistema deberá mover el personaje hacia esa dirección.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 47. RQF-21 Mover jugador a la izquierda

Id y Nombre	RQF-22 Mover jugador hacia arriba
Versión	v1
Descripción	El sistema deberá poder detectar la posición del joystick, tras esto, si la dirección es arriba, el sistema deberá mover el personaje hacia esa dirección.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 48. RQF-22 Mover Jugador hacia arriba

Id y Nombre	RQF-23 Mover jugador hacia abajo
Versión	v1
Descripción	El sistema deberá poder detectar la posición del joystick, tras esto, si la dirección es abajo, el sistema deberá mover el personaje hacia esa dirección.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 49. RQF-23 Mover jugador abajo

Id y Nombre	RQF-24 Mover jugador diagonal superior derecha
Versión	v1
Descripción	El sistema deberá poder detectar la posición del joystick, tras esto, si la dirección es arriba a la derecha, el sistema deberá mover el personaje hacia esa dirección.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 50. RQF-24 Mover jugador diagonal superior derecha

Id y Nombre	RQF-25 Mover jugador diagonal superior izquierda
Versión	v1
Descripción	El sistema deberá poder detectar la posición del joystick, tras esto, si la dirección es arriba a la izquierda, el sistema deberá mover el personaje hacia esa dirección.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 51. RQF-25 Mover jugador diagonal superior izquierda

Id y Nombre	RQF-26 Mover jugador diagonal inferior derecha
Versión	v1
Descripción	El sistema deberá poder detectar la posición del joystick, tras esto, si la dirección es abajo a la derecha, el sistema deberá mover el personaje hacia esa dirección.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 52. RQF-26 Mover jugador diagonal inferior derecha

Id y Nombre	RQF-27 Mover jugador diagonal inferior izquierda
Versión	v1
Descripción	El sistema deberá poder detectar la posición del joystick, tras esto, si la dirección es abajo a la izquierda, el sistema deberá mover el personaje hacia esa dirección.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 53. RQF-27 Mover jugador diagonal inferior izquierda

Id y Nombre	RQF-28 Vida de nuestro personaje
Versión	v1
Descripción	El sistema deberá poder gestionar la vida de nuestro jugador cuando sea alcanzado por algún enemigo.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 54. RQF-28 Vida de nuestro personaje

Id y Nombre	RQF-29 Atacar
Versión	v1
Descripción	El sistema deberá poder hacer que el jugador pueda atacar, hiriendo a los enemigos.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 55. RQF-29 Atacar

Id y Nombre	RQF-30 Morir
Versión	v1
Descripción	El sistema deberá poder detectar cuando la vida de nuestro personaje llega a cero, en dicho momento e personaje morirá.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 56. RQF-30 Morir

Id y Nombre	RQF-31 Recibir daño
Versión	v1
Descripción	El sistema deberá poder detectar cuando nuestro personaje sea alcanzado por un enemigo, restándole vida.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 57. RQF-31 Recibir daño

Id y Nombre	RQF-32 Realizar Daño
Versión	v1
Descripción	El sistema deberá poder detectar la colisión entre el arma del personaje principal y un enemigo y aplicar un daño.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 58. RQF-32 Realizar Daño

Id y Nombre	RQF-33 Generar partículas
Versión	v1
Descripción	El sistema deberá poder generar partículas cuando nuestro personaje haga daño a algún enemigo.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 59. RQF-33 Generar partículas

Id y Nombre	RQF-34 Destruir partículas
Versión	v1
Descripción	El sistema deberá poder destruir partículas después de su generación.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 60. RQF-34. Destruir partículas

Id y Nombre	RQF-35 Generar indicador numérico daño recibido
Versión	v1
Descripción	El sistema deberá poder generar un indicador numérico con el daño recibido.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 61. RQF-35 Generar indicador numérico daño recibido

Id y Nombre	RQF-36 Destruir indicador numérico daño recibido
Versión	v1
Descripción	El sistema deberá poder destruir el indicador de daño para que solo se muestre por un periodo corto de tiempo.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 62. RQF-36. Destruir indicador numérico daño recibido

Id y Nombre	RQF-37 Generar indicador numérico daño realizado
Versión	v1
Descripción	El sistema deberá poder generar un indicador numérico con el daño realizado.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 63. RQF-37. Generar Indicador numérico daño realizado

Id y Nombre	RQF-38 Destruir indicador numérico daño realizado
Versión	v1
Descripción	El sistema deberá poder destruir el indicador de daño realizado para que solo se muestre por un periodo corto de tiempo.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 64. RQF-38. Destruir Indicador numérico daño realizado

Id y Nombre	RQF-39 Generar animación cuando nuestro personaje sea alcanzado por un enemigo
Versión	v1
Descripción	El sistema deberá poder generar un cambio en el personaje cuando sea alcanzado por los enemigos
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	Será un parpadeo morado.

Tabla 65. RQF-39. Generar animación cuando nuestro personaje sea alcanzado por un enemigo

Id y Nombre	RQF-40 Zonas de dialogo
Versión	v1
Descripción	El sistema deberá poder detectar que el personaje se encuentra en una zona de dialogo.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	Dos tipos de zonas de dialogo: - Automáticas: Se activan en el momento en el que el personaje pasa por cierta zona. - Manuales: El personaje tiene que entrar en cierta zona, pero además, es necesario la interacción con alguna clase de objeto(por ejemplo, un cuadro).

Tabla 66. RQF-40. Zonas de dialogo

Id y Nombre	RQF-41 Mostrar zona de diálogo
Versión	v1
Descripción	El sistema deberá poder mostrar al jugador un mensaje o mensajes tras entrar y/o entrar y activar la zona de diálogo.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 67. RQF-41. Mostrar zona de diálogo

Id y Nombre	RQF-42 Cargar Nivel
Versión	v1
Descripción	El sistema deberá poder ir cargando los distintos niveles.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	En este caso solo habrá un nivel implementado.

Tabla 68. RQF-40 Cargar nivel

Id y Nombre	RQF-43 Limite Mapa
Versión	v1
Descripción	El sistema deberá poder gestionar como se comportara nuestro personaje al colisionar con las paredes de las salas, así como el comportamiento de las puertas.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 69. RQF-43. Limite Mapa

5.3.2 Requisitos no Funcionales

Id y Nombre	RQNF-01 Acceso Dispositivo
Versión	v1
Descripción	El sistema no necesitará permisos especiales para su instalación en el dispositivo, pero si necesitará acceder al almacenamiento del dispositivo móvil para poder instalarla.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 70. RQNF-01 Acceso al almacenamiento del dispositivo

Id y Nombre	RQNF-02 Almacenamiento Disponible
Versión	v1
Descripción	El dispositivo móvil, necesitara de 23.2 MB de almacenamiento para su instalación.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 71. RQNF-02 Almacenamiento mínimo

5.3.2.1 Requisitos de Rendimiento

Id y Nombre	RQNFR-01 Respuesta Rápida
Versión	v1
Descripción	El sistema deberá poder gestionar de manera eficiente el tiempo de respuesta en la interacción usuario-juego, dotándolo de una mejor experiencia al usuario debido a su rendimiento.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 72. RQNFR-02 Respuesta Rápida

5.3.2.2 Requisitos de Escalabilidad

Id y Nombre	RQNFE-01 Compatibilidad distintos dispositivos Android
Versión	v1
Descripción	La aplicación debe ser compatible independientemente del dispositivo, siempre y cuando se tenga Android como sistema operativo, sin importar su versión.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 73. RQNFE-01 Compatibilidad dispositivos Android

Id y Nombre	RQNFE-02 Compatibilidad distintas resoluciones pantalla de dispositivos
Versión	v1
Descripción	La aplicación debe ser compatible independientemente de la resolución de pantalla que posean los dispositivos móviles.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 74. RQNFE-02 Compatibilidad resoluciones dispositivos

5.3.2.3 Requisitos de Hardware

Id y Nombre	RQNFH-01 Dispositivo Android
Versión	v1
Descripción	El usuario deberá disponer de un dispositivo portátil Android.
Dependencias	
Prioridad	Alta
Frecuencia de uso	Alta
Otra información	

Tabla 75. RQNFH-01 Dispositivo Android

5.3.2.4 Requisitos de Internacionalización

Id y Nombre	RQNFI-01 Idioma
Versión	v1
Descripción	El Sistema en sus primeras versiones solo tendrá disponible de idioma el español.
Dependencias	
Prioridad	Media
Frecuencia de uso	Alta
Otra información	Para versiones más avanzadas se piensa incluir también la versión en inglés del juego.

Tabla 76. RQNFI-01 Idioma

5.4 Casos de uso

En esta sección se describe el comportamiento entre el sistema y los actores que harán uso de él.

Dado que es un videojuego de un jugador para móvil, sólo habrá un actor que es el mismo jugador.

5.4.1 Requisitos Diagrama de Casos de Uso

El **Diagrama3** especifica los Casos de Uso del videojuego implementando los objetivos principales.

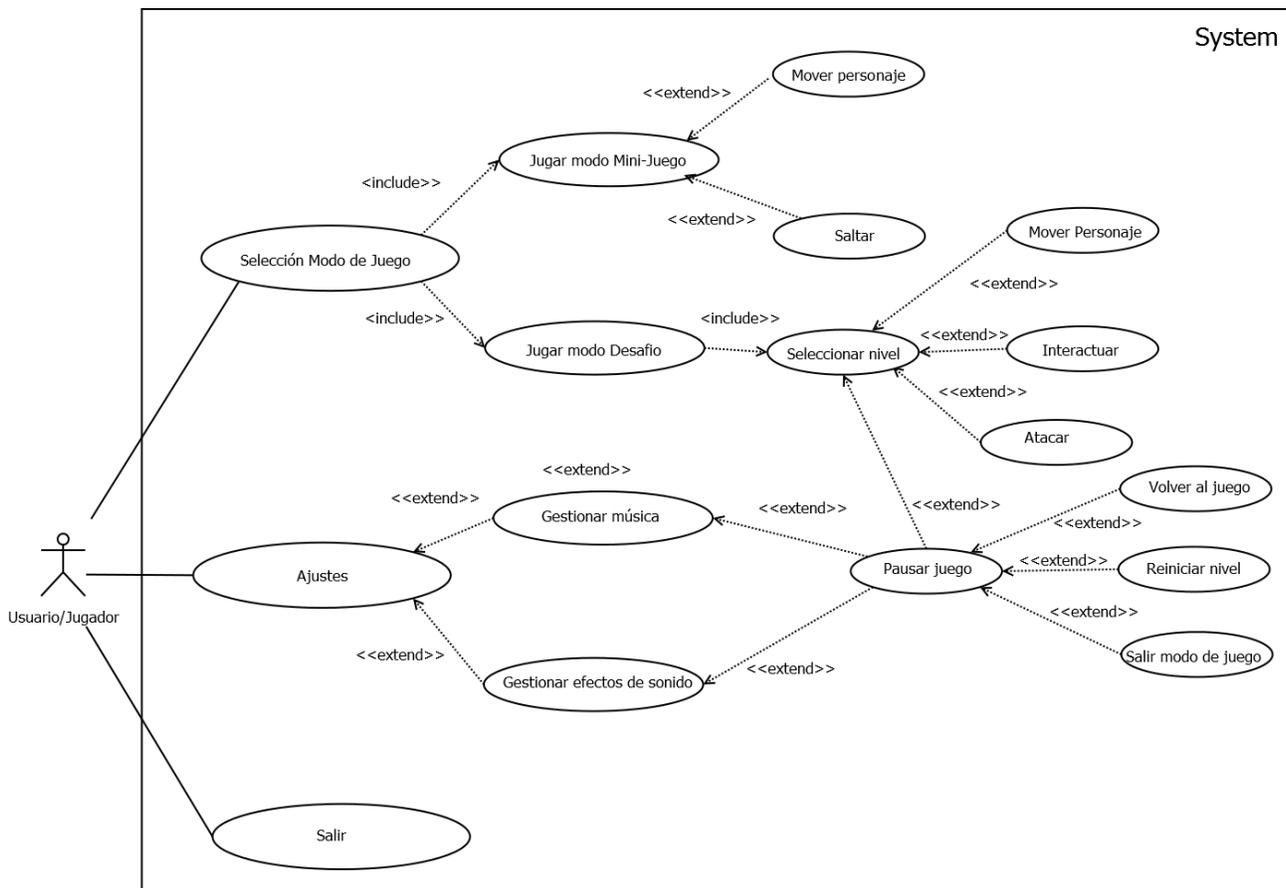


Diagrama 1. Modelo de Casos de Uso

5.4.2 Especificación de Casos de Uso

ID y Nombre	CU-01 Selección Juego	
Actor	Usuario/Jugador	
Descripción	Se muestra en pantalla un menú el cuál nos mostrará los distintos modos de juego a los que el usuario puede acceder, además de las opciones salir y configuración.	
Disparador		
Dependencias		
Precondiciones	Estar ubicado en la pantalla del menú principal del juego	
Flujo normal	Paso	Acción
	1	
Flujo alternativo		
Postcondiciones		
Excepciones		
Prioridad	Alta	
Frecuencia de uso	Alta	
Otra información		

Tabla 77. CU-01 Selección juego

ID y Nombre	CU-02 Jugar Mini-Juego	
Actor	Usuario/Jugador	
Descripción	Comienza la aventura “plataformera”, se inicia el juego.	
Disparador	El jugador deberá presionar el botón táctil: Play	
Dependencias		
Precondiciones	Estar ubicado en la pantalla del menú principal del juego	
Flujo normal	Paso	Acción
	1	Presionar el botón “PLAY” de la pantalla
Flujo alternativo		
Postcondiciones		
Excepciones		
Prioridad	Alta	
Frecuencia de uso	Alta	
Otra información		

Tabla 78. CU-02 Jugar Mini-juego

ID y Nombre	CU-03 Mover Personaje Mini-Juego	
Actor	Usuario/Jugador	
Descripción	El personaje se mueve en la dirección derecha de forma constante y sin parar.	
Disparador	El jugador toca la pantalla para iniciar el movimiento de nuestro jugador	
Dependencias	CU-01	
Precondiciones	Se debe haber seleccionado dicho modo de juego en el menú principal	
Flujo normal	Paso	Acción
	1	El jugador deberá tocar la pantalla táctil para iniciar el movimiento.
Flujo alternativo		
Postcondiciones		
Excepciones		
Prioridad	Muy Alta	
Frecuencia de uso	Muy Alta	
Otra información		

Tabla 79. CU-03 Mover personaje Mini-juego

ID y Nombre	CU-04 Saltar Mini-Juego	
Actor	Usuario/Jugador	
Descripción	El personaje realiza un pequeño salto	
Disparador	El jugador toca la pantalla táctil, en cualquier punto, haciendo saltar a personaje.	
Dependencias	CU-01	
Precondiciones	Estar ubicado en la pantalla del menú principal del juego El personaje tiene que tener saltos disponibles	
Flujo normal	Paso	Acción
	1	El jugador toca la pantalla táctil haciendo saltar al personaje.
Flujo alternativo		
Postcondiciones		
Excepciones	El personaje no dispone de saltos.	
Prioridad	Alta	
Frecuencia de uso	Alta	
Otra información	El número máximo de saltos consecutivos es: 2	

Tabla 80. CU-04 Saltar Mini-juego

ID y Nombre	CU-05 Jugar Modo Desafío	
Actor	Usuario/Jugador	
Descripción	Comienza el desafío del “templo oscuro”.	
Disparador	El jugador deberá presionar el botón táctil: Play	
Dependencias		
Precondiciones	Estar ubicado en la pantalla del menú principal del juego En el panel que aparece al presionar el botón de selección del juego, seleccionar “comenzar desafío”	
Flujo normal	Paso	Acción
	1	Presionar el botón “Desafío”
Flujo alternativo		
Postcondiciones		
Excepciones		
Prioridad	Alta	
Frecuencia de uso	Alta	
Otra información		

Tabla 81. CU-05 Jugar Modo Desafío

ID y Nombre	CU-06 Selección nivel	
Actor	Usuario/Jugador	
Descripción	Se llevara a cabo el proceso de carga del nivel seleccionado	
Disparador	El jugador selecciona un nivel en el menú de “selección nivel”	
Dependencias	CU-05	
Precondiciones	Estar ubicado en la pantalla de selección de nivel del modo desafío.	
Flujo normal	Paso	Acción
	1	Seleccionar nivel.
Flujo alternativo		
Postcondiciones		
Excepciones	Nivel seleccionado bloqueado	
Prioridad	Alta	
Frecuencia de uso	Alta	
Otra información		

Tabla 82. CU-06 Selección Nivel

ID y Nombre	CU-07 Mover personaje Desafío	
Actor	Usuario/Jugador	
Descripción	El personaje se mueve en la dirección hacia donde presionemos el joystick táctil.	
Disparador	El jugador mueve el joystick táctil	
Dependencias	CU-01, CU-06	
Precondiciones	Se debe haber seleccionado dicho modo de juego en el menú principal Se debe haber seleccionado un nivel	
Flujo normal	Paso	Acción
	1	El jugador deberá tocar la pantalla táctil para iniciar el movimiento.
Flujo alternativo		
Postcondiciones		
Excepciones		
Prioridad	Alta	
Frecuencia de uso	Muy Alta	
Otra información		

Tabla 83. CU-07 Mover personaje Desafío

ID y Nombre	CU-08 Atacar Desafío	
Actor	Usuario/Jugador	
Descripción	El personaje mueve su bastón hacia adelante, haciendo daño en el caso que hubiera un enemigo.	
Disparador	El jugador presiona el botón táctil circular rojo.	
Dependencias	CU-01, CU-06	
Precondiciones	Se debe haber seleccionado dicho modo de juego en el menú principal Se debe haber seleccionado un nivel	
Flujo normal	Paso	Acción
	1	Presionar el botón táctil circular rojo para atacar.
Flujo alternativo		
Postcondiciones		
Excepciones		
Prioridad	Alta	
Frecuencia de uso	Muy Alta	
Otra información		

Tabla 84. CU-08 Atacar Desafío

ID y Nombre	CU-09 Interactuar Desafío	
Actor	Usuario/Jugador	
Descripción	El personaje interactuara con NPC´s, ítems u objeto de decoración del escenario/sala en la que se encuentre.	
Disparador	El jugador presiona el botón táctil circular morado.	
Dependencias	CU-01, CU-06	
Precondiciones	Se debe haber seleccionado dicho modo de juego en el menú principal Se debe haber seleccionado un nivel	
Flujo normal	Paso	Acción
	1	Presionar el botón táctil circular morado para interactuar.
Flujo alternativo		
Postcondiciones		
Excepciones	Si se tratara de una misión/quest dada por un npc y esta estuviera ya realizada, el personaje no podrá interactuar mas si esté no tiene mas misiones para el. Si la misión/quest se tratara de la recogida de un ítem, no se podrá volver a interactuar con él ya que este desaparecerá.	
Prioridad	Alta	
Frecuencia de uso	Muy Alta	
Otra información		

Tabla 85. CU-09 Interactuar Desafío

ID y Nombre	CU-10 Menú pausa Desafío	
Actor	Usuario/Jugador	
Descripción	Pausa el juego mostrando el menú de PAUSA.	
Disparador	El jugador presiona el botón superior-derecho MENÚ	
Dependencias	CU-01, CU-06	
Precondiciones	Se debe haber seleccionado el modo de juego Desafío en el menú principal Se debe haber seleccionado un nivel	
Flujo normal	Paso	Acción
	1	Presionar el botón táctil ubicado en la parte superior-derecha de la pantalla (Menú Pausa).
Flujo alternativo		
Postcondiciones		
Excepciones	El jugador se encuentra en la pantalla GameOver	
Prioridad	Alta	
Frecuencia de uso	Muy Alta	
Otra información		

Tabla 86. CU-10 Menú pausa Desafío

ID y Nombre		CU-11 Reiniciar Nivel Desafío	
Actor	Usuario/Jugador		
Descripción	Se reinicia el nivel actual en el que se encuentre el jugador, perdiendo el progreso alcanzado hasta el momento.		
Disparador	El jugador presiona el botón “reiniciar”, en el menú de pausa.		
Dependencias	CU-01, CU-06, CU-10		
Precondiciones	Se debe haber seleccionado dicho modo de juego en el menú principal Se debe haber seleccionado un nivel Se debe estar ubicado en el menú de pausa		
Flujo normal	Paso	Acción	
	1	Se reinicia el nivel actual.	
Flujo alternativo			
Postcondiciones			
Excepciones			
Prioridad	Alta		
Frecuencia de uso	Media		
Otra información			

Tabla 87. CU-11 Reiniciar Nivel Desafío

ID y Nombre		CU-12 Cerrar menú pausa Desafío	
Actor	Usuario/Jugador		
Descripción	Vuelve al juego, ocultando el menú de pausa.		
Disparador	El jugador presiona el botón atrás.		
Dependencias	CU-01, CU-06		
Precondiciones	Se debe haber seleccionado modo de juego Desafío en el menú principal Se debe haber seleccionado un nivel Se debe estar ubicado en el menú de pausa		
Flujo normal	Paso	Acción	
	1	Presionar el botón táctil ubicado en la parte inferior-derecha de la pantalla (Menú Pausa).	
Flujo alternativo			
Postcondiciones			
Excepciones	El jugador se encuentra en la pantalla GameOver		
Prioridad	Alta		
Frecuencia de uso	Media		
Otra información			

Tabla 88. CU-12 Cerrar menu pausa Desafío

ID y Nombre	CU-13 Salir a menú selección juego Desafío	
Actor	Usuario/Jugador	
Descripción	Vuelve a la pantalla de menú principal, selección de juego.	
Disparador	El jugador presiona el botón “Salir”.	
Dependencias	CU-01, CU-06	
Precondiciones	Se debe haber seleccionado dicho modo de juego en el menú principal Se debe haber seleccionado un nivel Se debe estar ubicado en el menú de pausa	
Flujo normal	Paso	Acción
	1	Presionar el botón táctil “Salir”.
Flujo alternativo		
Postcondiciones		
Excepciones		
Prioridad	Alta	
Frecuencia de uso	Media	
Otra información		

Tabla 89. CU-13 Salir a menu selección juego Desafío

ID y Nombre	CU-14 Configuración	
Actor	Usuario/Jugador	
Descripción	Vuelve a la pantalla de menú principal, selección de juego.	
Disparador	El jugador presiona el botón “Configuración”.	
Dependencias		
Precondiciones		
Flujo normal	Paso	Acción
	1	Presionar el botón táctil “Configuración”.
Flujo alternativo	Presionar botón configuración desde menú Pausa Desafío.	
Postcondiciones		
Excepciones		
Prioridad	Alta	
Frecuencia de uso	Media	
Otra información		

Tabla 90. CU-14 Configuración

ID y Nombre	CU-15 Salir	
Actor	Usuario/Jugador	
Descripción	Sale del juego	
Disparador	El jugador presiona el botón “Salir”.	
Dependencias		
Precondiciones	Se debe encontrar en el menú de selección de juego –HomeScene-	
Flujo normal	Paso	Acción
	1	Presionar el botón táctil “Salir”.
Flujo alternativo		
Postcondiciones		
Excepciones		
Prioridad	Alta	
Frecuencia de uso	Media	
Otra información		

Tabla 91. CU-15 Salir

ID y Nombre	CU-16 Gestionar música	
Actor	Usuario/Jugador	
Descripción	Activar/Desactivar la música de fondo del juego	
Disparador		
Dependencias	CU-01, CU-06	
Precondiciones	Se debe haber entrado en el panel del menú de configuración ya sea desde la escena de selección de juego o desde el modo de juego –Desafío-	
Flujo normal	Paso	Acción
	1	Presionando botón configuracion
Flujo alternativo	Paso	Acción
	1	Presionando botón configuración tras pulsar botón Menu Pausa en modo –Desafío-
Postcondiciones		
Excepciones		
Prioridad	Alta	
Frecuencia de uso	Media	
Otra información		

Tabla 92. CU-16 Gestionar música

ID y Nombre	CU-17 Gestionar efectos de sonido	
Actor	Usuario/Jugador	
Descripción	Activar/Desactivar los efectos de sonido del juego.	
Disparador		
Dependencias	CU-01, CU-06	
Precondiciones	Se debe haber entrado en el panel del menú de configuración ya sea desde la escena de selección de juego o desde el modo de juego – Desafío-	
Flujo normal	Paso	Acción
	1	Presionar el botón táctil “Salir”.
Flujo alternativo	Paso	Acción
	1	Presionando botón configuración tras pulsar botón Menu Pausa en modo –Desafío-
Postcondiciones		
Excepciones		
Prioridad	Media	
Frecuencia de uso	Media	
Otra información	No implementado, en un futuro se encontrará en el mismo panel de configuración.	

Tabla 93. CU-17 Gestión de efectos de sonido

5.5 Árbol de características

En esta sección se llevará a cabo la realización de un Árbol de Características sencillo, para su mejor interpretación y tendrá como objetivo el mostrar las características principales del sistema.

- CP1. Seleccionar Modo de Juego.**
- CP3. Jugar Mini-juego Plataforma.**
 - CP5. Mover Personaje**
 - CP8.** Mover al personaje hacia la derecha de manera continua
 - CP9.** Hacer que el personaje salte
 - CP10.** Hacer que el personaje salte una vez extra.
 - CP4. Jugar Modo Desafío.**
 - CP6. Mover al personaje**
 - CP11.** Mover al personaje hacia a dirección: izquierda.
 - CP12.** Mover al personaje hacia a dirección: derecha.
 - CP13.** Mover al personaje hacia a dirección: abajo.
 - CP14.** Mover al personaje hacia a dirección: arriba.
 - CP15.** Mover al personaje hacia a dirección: diagonal superior derecha.
 - CP16.** Mover al personaje hacia a dirección: diagonal superior izquierda.
 - CP17.** Mover al personaje hacia a dirección: diagonal inferior derecha.
 - CP18.** Mover al personaje hacia a dirección: diagonal inferior izquierda.
 - CP19.** hHacer que el personaje ataque
 - CP20.** Hacer que el personaje interactué
 - CP7. Pausar juego.**
 - CP21.** Cerrar menu pausa.
 - CP21.** Reiniciar el nivel.
 - CP22.** Salir del juego.
- CP2. Ajustes.**
 - CP23.** Gestionar música.
 - CP24.** Gestionar sonido.

Tabla 94. Características principales

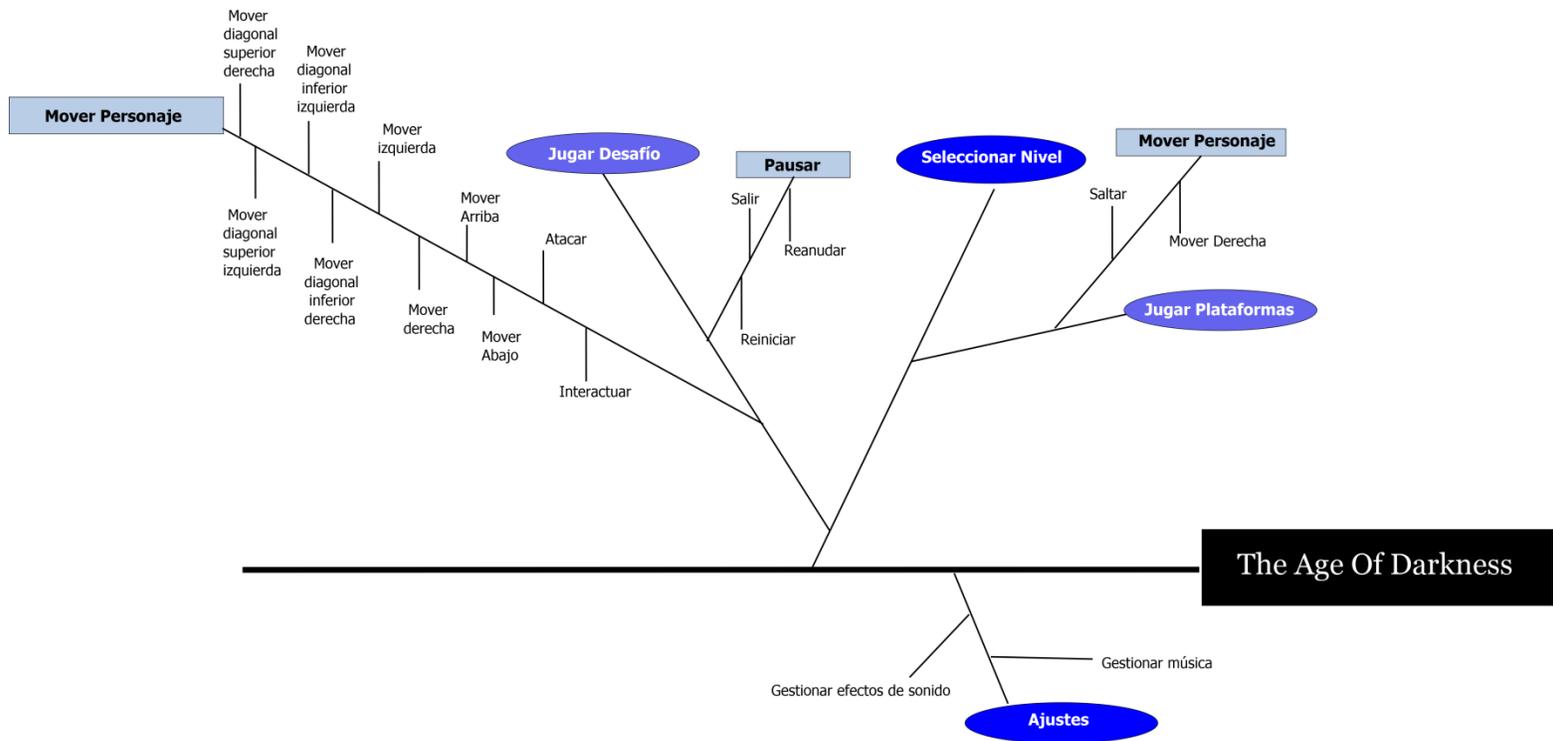


Figura 7. Árbol de Características

5.6 Diagramas de secuencia

En esta sección se muestra la interacción entre los distintos objetos a la hora de llevar a cabo las funcionalidades principales del juego.

• Saltar (Modo Plataforma)

Diagrama: Secuencia para hacer saltar a nuestro personaje

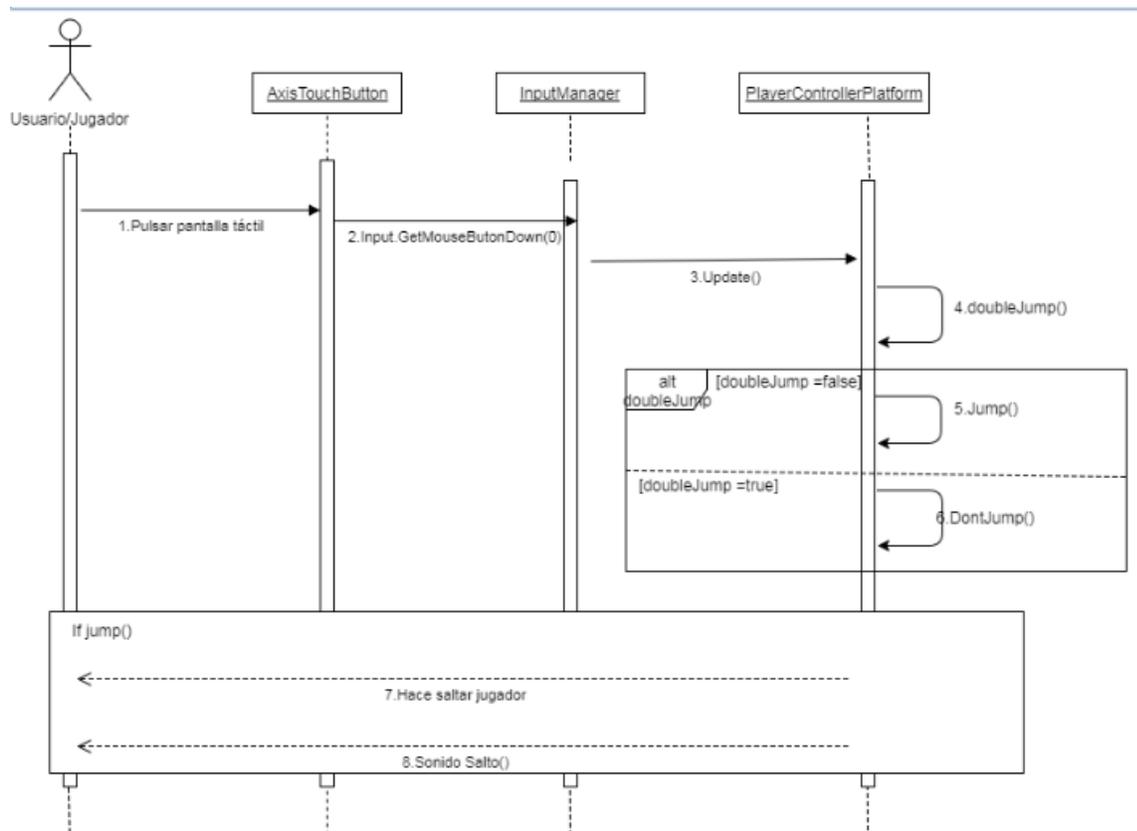


Diagrama 2. Diagrama de secuencia para que nuestro personaje salte

• Crystal (Modo Plataforma)

Diagrama: Secuencia para recoger un ítem del nivel.

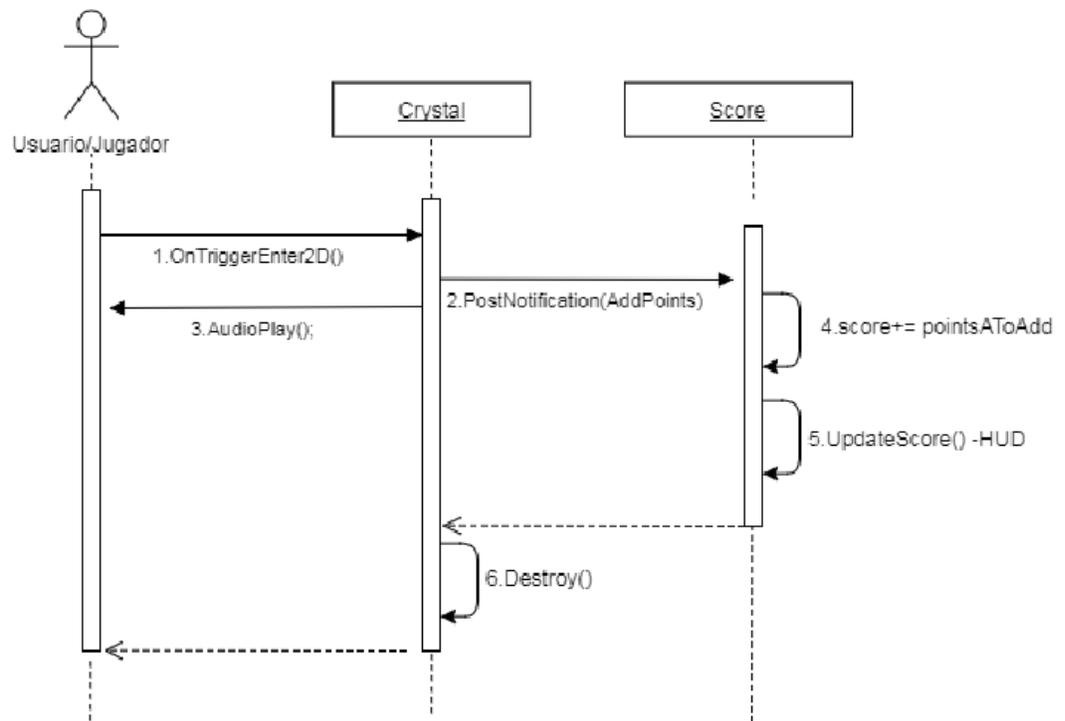


Diagrama 3 Diagrama de secuencia para coger un ítem del juego

● MenuPausa (Modo Desafío)

Diagrama: Secuencia para acceder al menú de pausa.

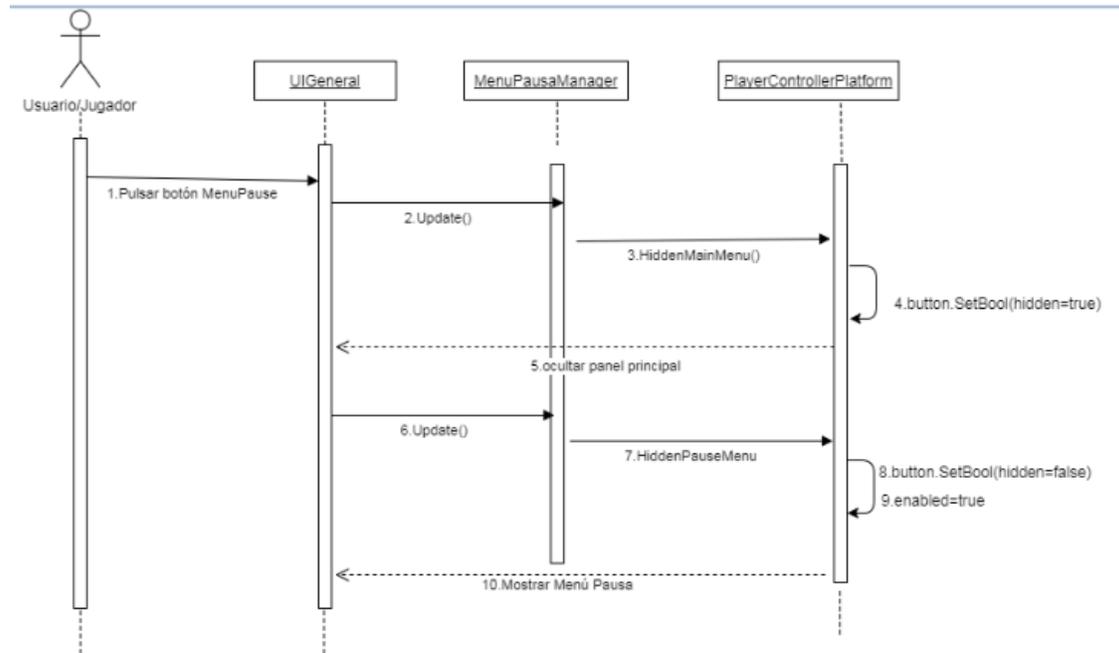


Diagrama 4 Diagrama de secuencia para abrir el menú “pausa” del juego

● MoverPersonaje (Modo Desafío)

Diagrama: Secuencia para mover a nuestro personaje

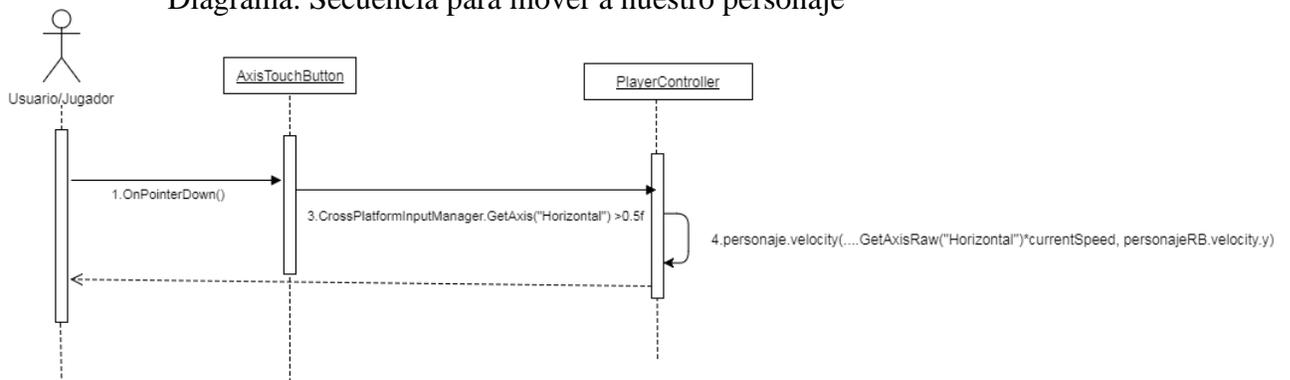


Diagrama 5. Diagrama de secuencia para mover al personaje hacia la derecha(Desafío)

• Atacar Personaje (Modo Desafío)

Diagrama: Secuencia para hacer que nuestro personaje ataque.

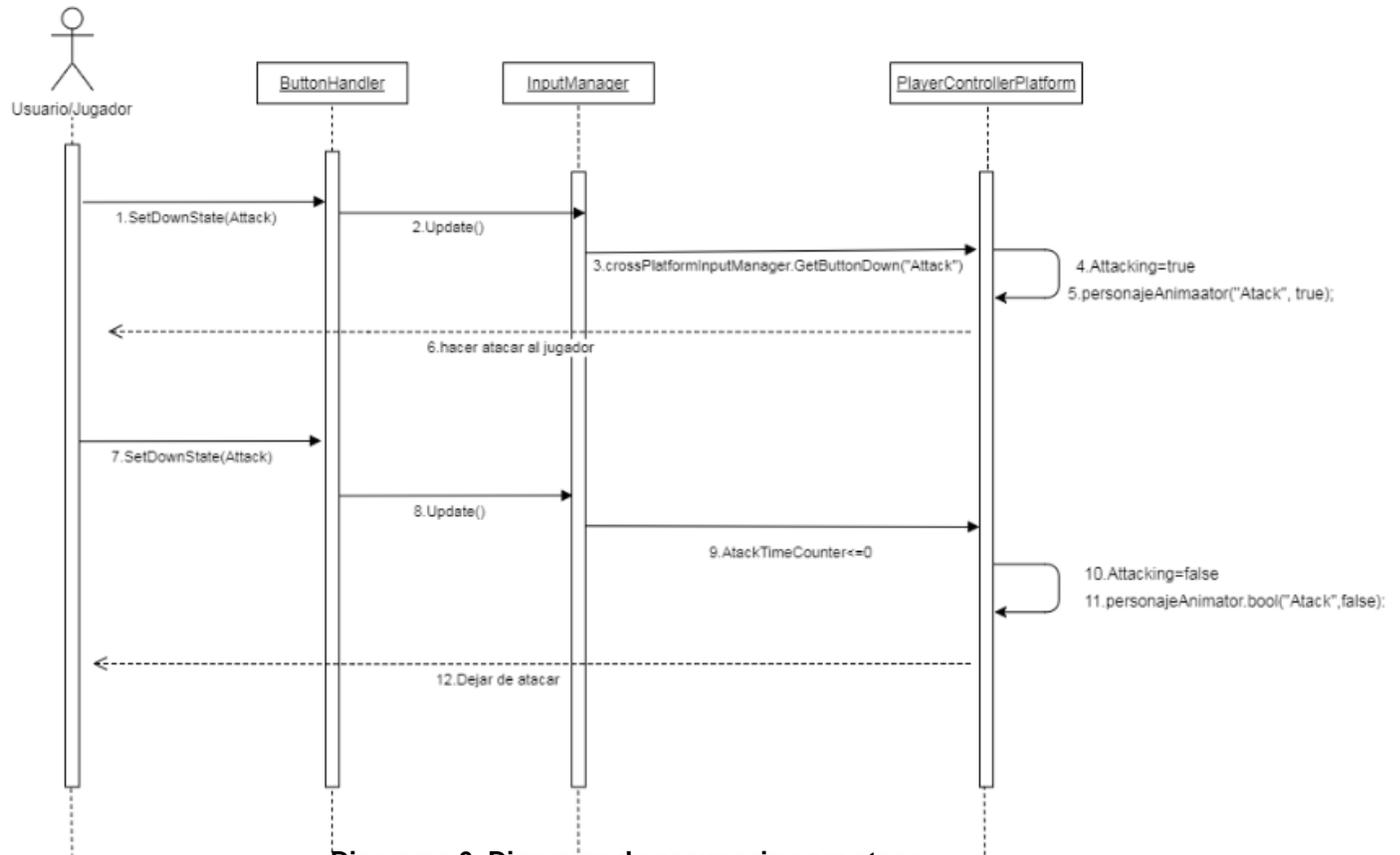


Diagrama 6. Diagrama de secuencia para atacar

Capítulo 6

Diseño

En este capítulo se va a exponer la arquitectura lógica y física del videojuego y la fase de diseño, donde se explica cómo se ha diseñado cada componente del videojuego, y para qué. También se expone la Interfaz de Usuario (IU).

6.1 Arquitectura lógica

En esta sección se muestra las diferentes capas en las que la arquitectura lógica de la aplicación se divide.

La arquitectura lógica de la aplicación se divide en diferentes capas:

- 1. Capa de presentación:** Es la encargada de mostrar el sistema al usuario. Se comunica con la *capa de negocio*.
- 2. Capa de negocio:** Contiene los scripts con las funcionalidades del juego, y las entidades de negocio, que se encargan del intercambio de datos entre capas. Recibe las solicitudes de la *capa de presentación*.
- 3. Capa de datos:** En este caso no se encuentra en el proyecto esta capa debido a que no hay, no se hace uso de bases de datos externas.
- 4. Capa de componentes comunes:** componentes a los que puede accederse desde cualquier parte de la aplicación, como por ejemplo la configuración del sonido del juego.

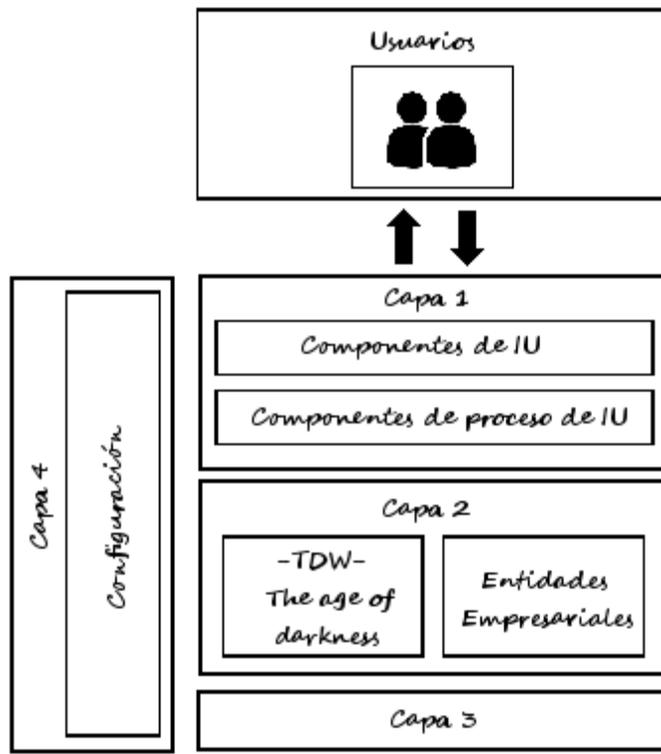


Figura 8 Arquitectura lógica

6.2 Arquitectura física

En esta sección se muestra una breve descripción sobre que elemento/s físicos intervienen.

El dispositivo al que ira dirigido sera a un dispositivo móvil y sera el unico elemento físico necesario para la ejecución de la aplicación

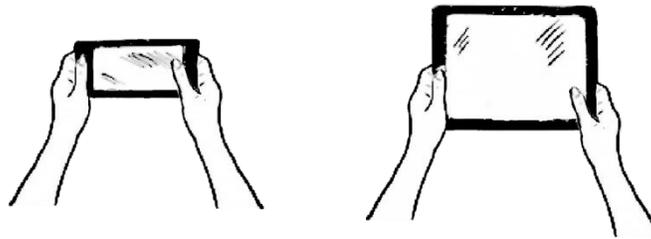


Figura 9. Arquitectura física

6.3 Diagramas de clases de diseño

En esta sección se llevara a cabo un presentación sobre las clases que conforman el sistema y como se relacionan, permitiéndonos conocer la estrucra del sistema.

A modo de introducción se mostrará la estructura de Unity3D.

Estructura Unity3D y sus clases:

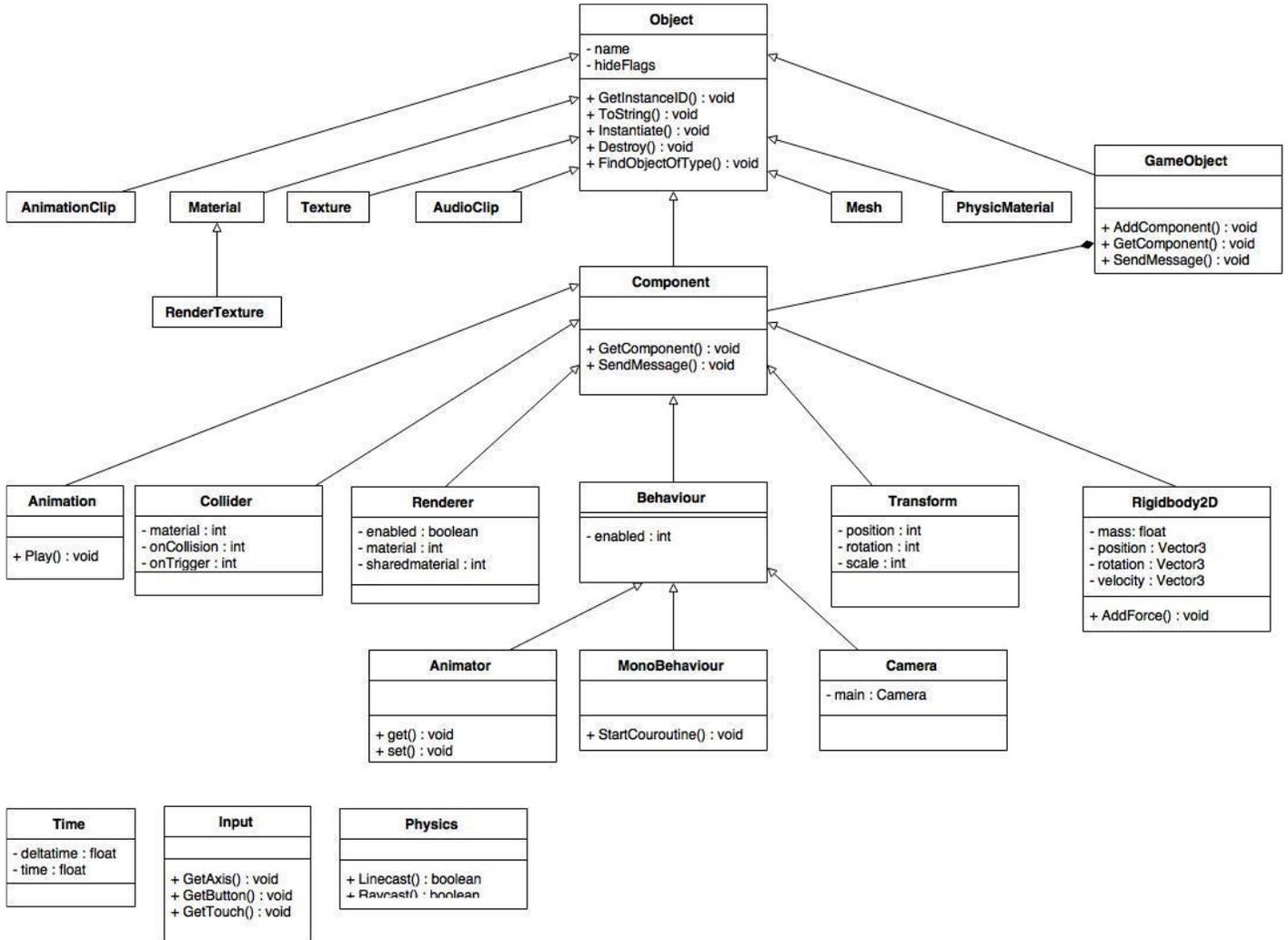


Diagrama 7. Diagrama de clases de la estructura de Unity3D

Como nos muestra el diagrama anterior, se puede “leer” fácilmente lo que ya se ha ido comentando a lo largo del proyecto, es decir, que Unity3D se basa en Objetos, los cuales poseen distintos componentes y, que para ser ejecutado en Unity3D debe heredar de la clase Mono-Behaviour

Tras esta breve introducción se va a pasar a explicar ahora las clases del proyecto, estás, debido a que el alcance del proyecto es muy amplio, se dividirá en varios diagramas de clases a los cuales sólo se les añadirán atributos y funciones principales

● **Controladores de juego: Diagrama**

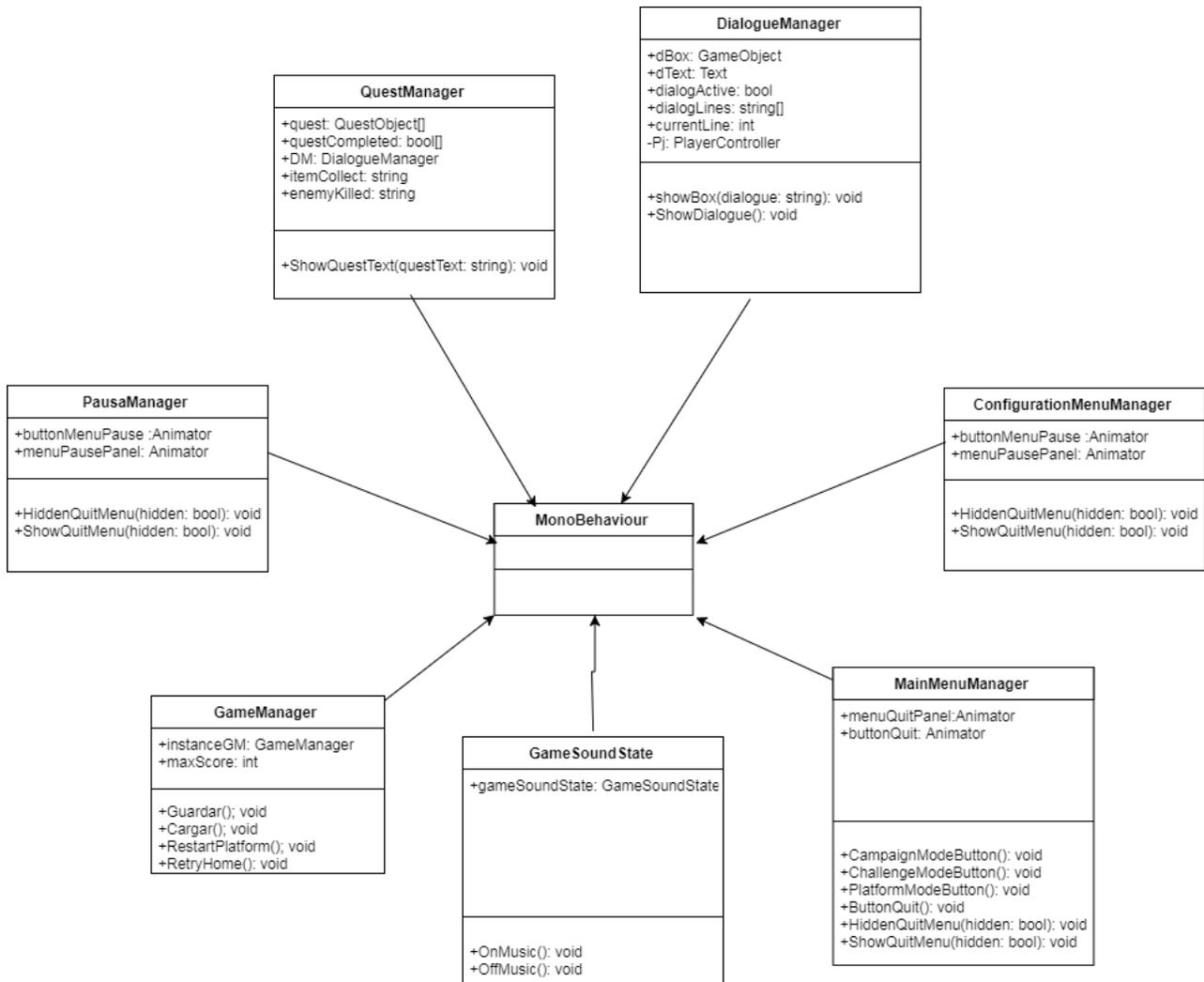


Diagrama 8. Diagrama de clases de los controladores del juego (General)

● **Personaje del juego: Diagrama**

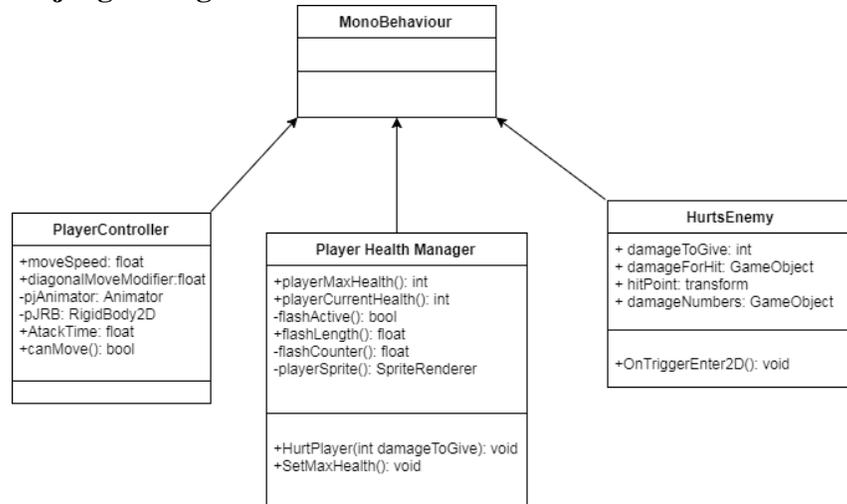


Diagrama 9. Diagrama de clases del personaje del juego (Desafío)

● **Enemies: Diagrama**

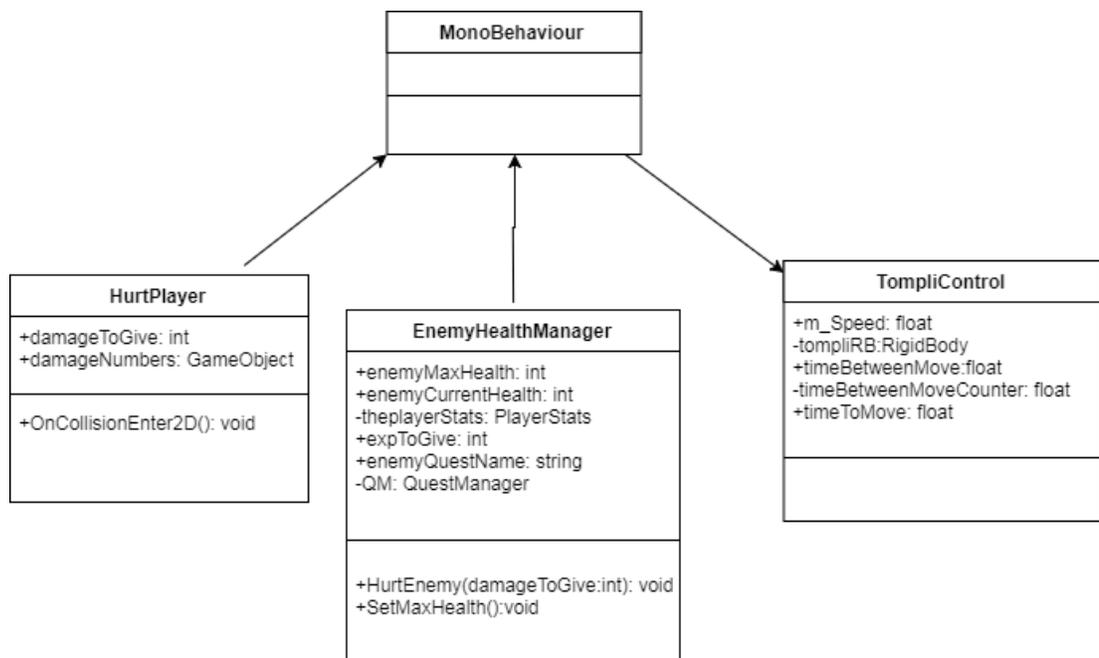


Diagrama 10. Diagrama de clases de los enemigos (Desafío)

- **Objetos del juego: Diagrama**

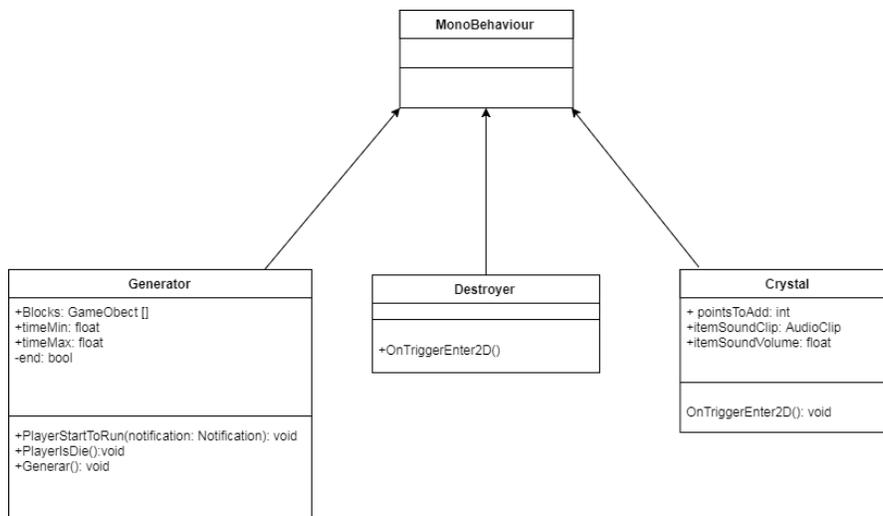


Diagrama 11. Diagrama de clases de los objetos del modo de juego plataformas

6.4 Diagramas de estados

En esta sección se describirá brevemente por qué estados podrá pasar el sistema.

- **Personaje:** El personaje del juego podrá pasar por distintos estados dependiendo de la acción que este hacienda en un momento determinado.
- **Menús:** Muchos de los menús que nos encontramos a lo largo del videojuego podrán estar “escondidos o no”.
- **Música:** La música del juego tiene el estado de activada y desactivada.

Personaje

Modo Plataforma

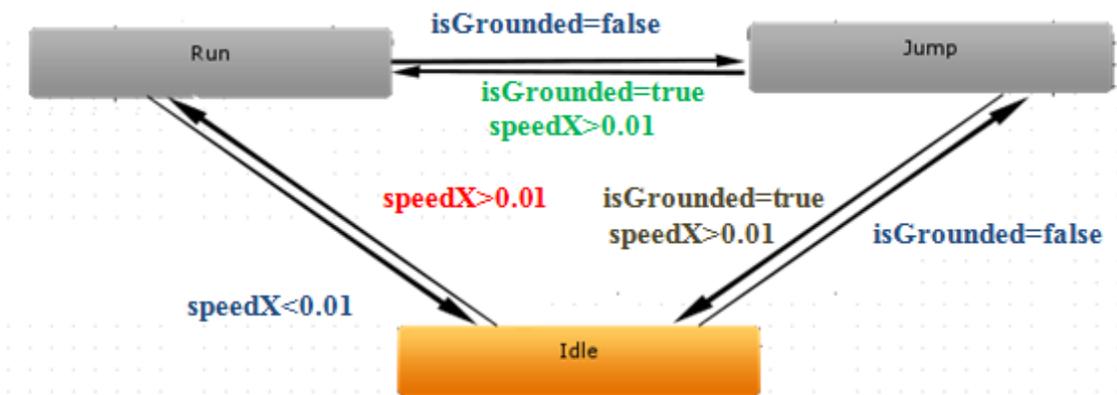
- **Idle:** Se trata de la animación por defecto del personaje, cuando no se encuentre en movimiento realizara dicha animación, la cuál dara sensación de estar parado, esperando para empezazar el nivel.
- **run:** cuando el jugador presione la pantalla táctil del dispositivo, se actualizara el estado de nuestro personaje activando dicha animación la cual le dará aspecto de estar corriendo,

esta animación se vera afectada si nuestro personaje salta.

- **Jump:** cuando el jugador presione la pantalla táctil del dispositivo, se actualizará el estado de nuestro personaje y, este saltará.

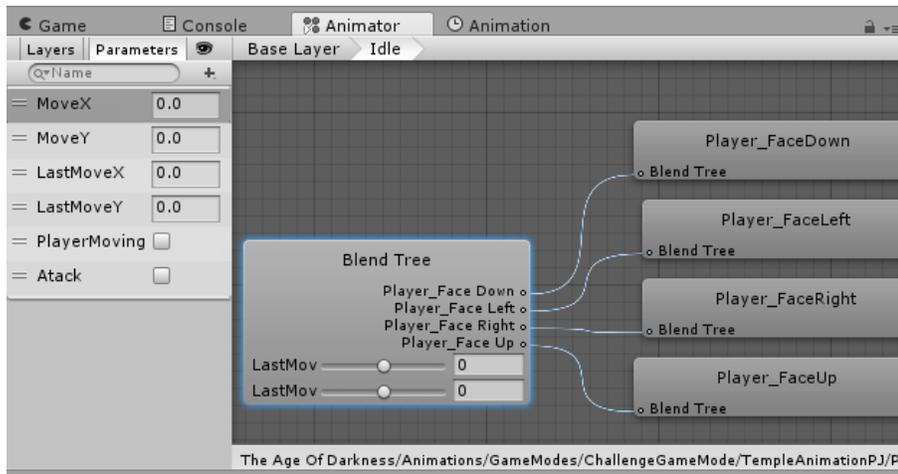
Jump Platform: Si se encuentra en una paltafoma saltará normalmente

Jump Jump: Si se trata de un segundo salto, nuestro personaje saltará normalmente, de llevar ya los dos saltos gastados nuestro personaje no podra saltar más por lo que no se llevara a cabo de nuevo la animación. (El personaje ene l aire solo podrá saltar una vez)



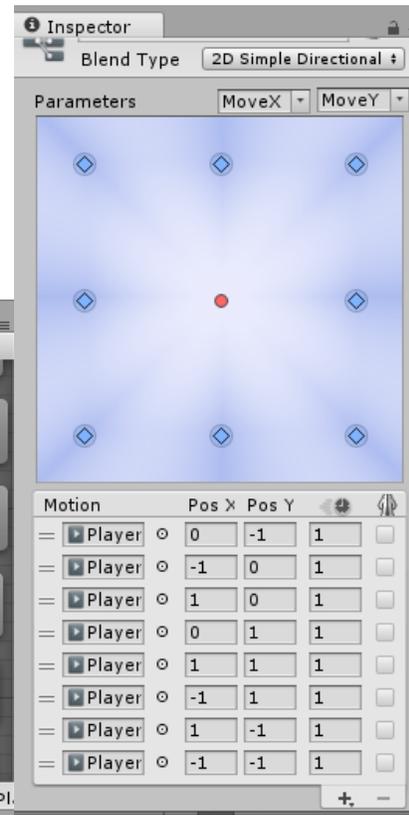
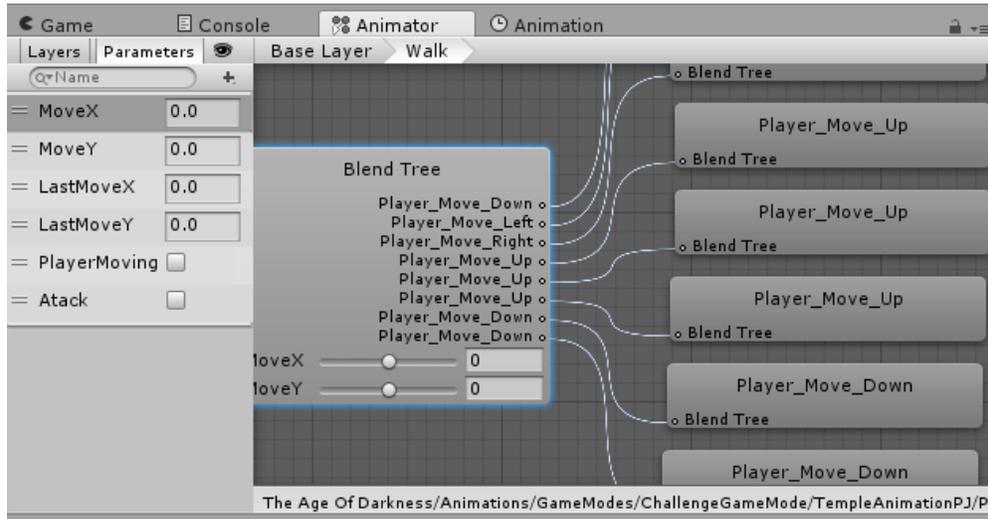
Modo Desafio

- **Idle:** Se trata de la animación por defecto del personaje, cuando no se encuentre en movimiento realizará dicha animación, la cuál dara sensación de estar parado, cabe destacar, que como en este modo de juego, internamente se deben controlar muchos estados, se hará uso de un **blend tree**⁴ para cada estado de animación(IdleDown/IdleUP/IdleRight/IdleLeft)

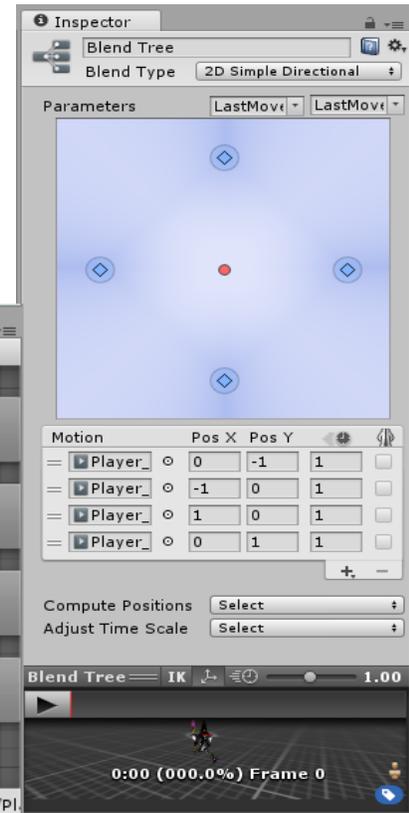
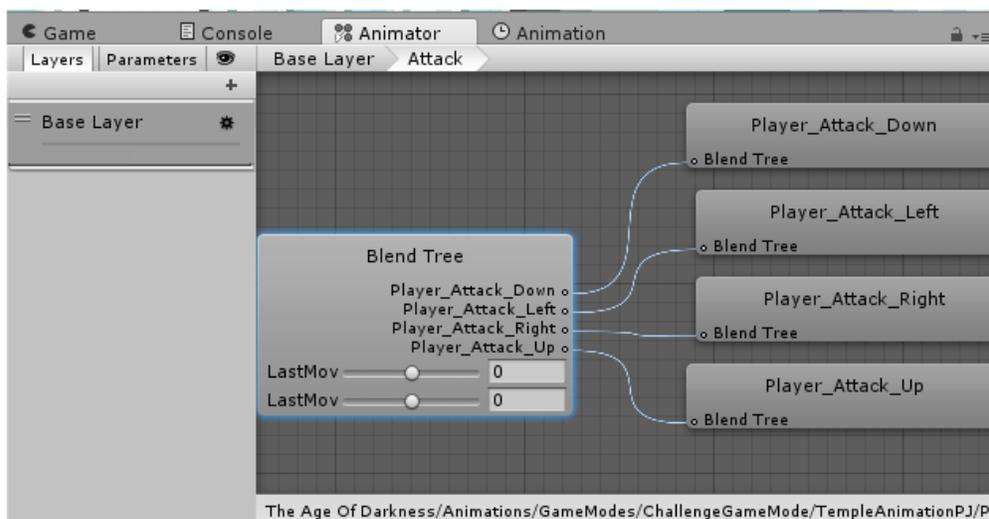


⁴ Blend Trees: Se utiliza para permitir múltiples animaciones.

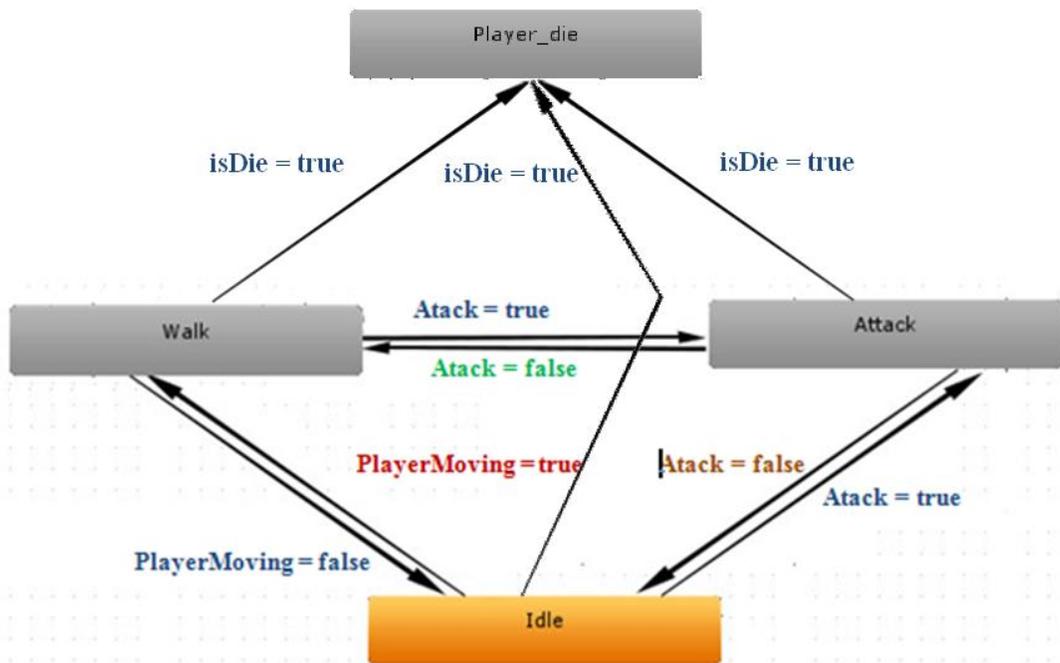
- **walk**: cuando el jugador presione y mueva el joystick , se dotará de movimiento al personaje “ifrahul”, utilizando como se ha descrito anteriormente un **Blend Tree** para la elección de la animación, con la diferencia que en este caso se controlarán cuatro estados mas, éstos serán las diagonales.



- **Attack**: cuando el jugador presione el botón táctil “ATACAR”, se pondrá en funcionamiento este **Blend Tree** que, a través de la posición en la que se encontraba el personaje elejirá una animación u otra.



Estados Generales:



Enemigo

Modo Desafío

Debido a que actualmente el único estado de movimiento disponible es “Random”, su diagrama será simple.

- **MovRandom:** Los enemigos que nos encontramos en el primer nivel tendrán un único estado el cual se iniciará cuando el usuario/jugador comience el nivel:

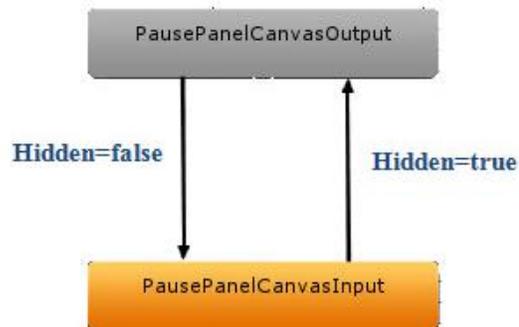


Este movimiento random tendrá velocidades distintas para cada enemigo para añadir un poco de “variedad”, a parte de la aleatoriedad de su movimiento.

Menús

Actualmente, algunos menús, entre los que podemos encontrar el menú de pausa del modo desafío o el menú de configuración (dónde se llevará a cabo el gestión de la música del juego) cabría destacar que tendrían también dos estados, estos serían Inpu/output, es decir, si el panel se encuentra fuera de de cámara o no.

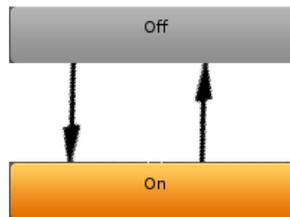
Ejemplo:



Música

Actualmente, la música del juego puede pasar por dos estados, encendido y apagado.

Ejemplo:



6.5 Historia

En este capítulo, se describirá la historia detrás de nuestro personaje, cabe destacar que aunque sea un juego “multi-juego” no constará de 3 historias, sino que la historia principal tratará sobre el modo de juego aun no implementado (“Campaña”) pero se irá complementando según vayamos avanzando en los otros dos modos de juego (“Plataforma” & “Desafío”) llegando a necesitar pasarnos determinando niveles para poder seguir avanzando en la campaña, o, para conseguir determinados objetos que ayudarán en la aventura.

La historia comenzó hace mucho tiempo, en un lejano lugar, cierto día en el que se iba a llevar a cabo un fenómeno llamado: “el eclipse de las 12 lunas”, éste, consistía en la alineación de 12 lunas y solo ocurría una vez cada 100 años, en ese momento, los espíritus de las lunas (llamados los 12 “sabios”) se reunían, éstos eran los encargados de mantener el orden en el mundo, es decir, de vigilar a “la luz” y “la oscuridad” para que no se descontrolasen (para mantener el equilibrio en el mundo.), pero esta vez fue diferente, fue el comienzo de este **Nuevo Mundo**.

En el momento del eclipse, un meteorito compuesto por un material el cual se descubrió después, llamado “cristanita”, colisiono contra la primera luna, colapsando y dando lugar a lo que luego se llamo “Yügen⁵ ikigai⁶”.

Ante este fenómeno, el sabio de la luna colisionada desapareció, dejando solo 11 sabios, por lo tanto debido a que les influía directamente decidieron no descansar esta vez y ver como evolucionaba.

Mucho tiempo después, nos encontramos con un mundo prospero, en el que, debido a las condiciones tan especiales en el surgimiento de dicho mundo, nos encontramos con la presencia de dos personificaciones muy potentes, las cuales, se originaron a partir del primer sabio.

De por sí, cada sabio, carece del “bien/mal” no están sujetos a esas leyes, eran “seres transcendentales” superiores a toda clase de sentimiento humano, pero, debido a que la colisión ocurrió el mismo día que el gran eclipse de las 12 lunas esto, hizo que el sabio perdiera su estado de Transcendencia desembocando en la aparición de la condición humana, el bien y el mal, la luz y la oscuridad.

Así pues, nos encontraríamos con tres tipos de fuerzas de poder: el lado de la luz (“el bien”), el lado de la oscuridad (“el mal”) y los 11 sabios. Estos últimos decidieron crear un pacto con las otras dos fuerzas, se crearía un pacto por el cual el uno solo podría convivir con la existencia del otro, y los 11 sabios, se encargarían de mantener ambos en la misma “proporción”, forjando la “idea” de no existir el bien sin el mal, aunque, cabe destacar que los 11 sabios siempre tuvie-

⁵ Yügen: Palabra japonesa que quiere decir profundo/misterioso y se refiere a un conocimiento del universo que evoca sentimientos emocionales que son inexplicablemente profundos y que es demasiado misterioso para las palabras.

⁶ Ikigai: Palabra japonesa que se refiere a la razón de vivir, eso que hace que te levantes cada día. Para encontrarla suele ser necesaria una profunda búsqueda interior.

ron cierta predilección por el lado de “luz” .

Estos, llegado cierto momento, se corrompen por el lado del “bien” el cuál desea acabar con el mal en el mundo y trazan un plan conjunto para, como no hay bien sin mal, conseguir sellarlo (no se le podría matar, de ahí viene que cada vez que el personaje muera se active una animación “Ritual-Serpiente” lo cual nos permitirá volver a reintentar el nivel(Desafío) o en modo campaña volveremos al ultimo punto de guardado/posada), es decir, intentar sellar el mal del mundo en un “cristal de hipolonita” un mineral muy escaso en el mundo y cuyo tratamieto, al tratarse de un mineral tan especial, solo puede ser llevado a cabo bajo la utilización de determinados artefactos. Su condición de mineral especial se debe a que dicho material se encontraba en el núcleo del meteorito originador de dicho mundo junto con la “cristanita”

Fue una batalla ajustada, puesto que toda clase de ser maligno, ya fueran clases malvadas débiles(D) o clases malvadas fuertes (A/S), transpararon casi todo su poder maligno a su amo para intentar salvarlo, transformandose asi ellos en una versión temprana de ellos mismos los cuales poseían practicamente un pulso maligno nulo.

Tras finalizar la batalla, el lado de la Luz junto con los 11 Sabios habian salido victoriosos, consiguiendo desterrar al mal en el cristal de hipolonita, esperando a ser despertado...

Estos seres, llamados “hollow/huecos”, pudieron esconderse “en las sombras” debido a su escaso pulso maligno, introduciendose dentro de la gente en forma de los 7 pecados capitales, para así poder coexistir sin ser detectados por el lado de “luz”, aguardando a poder vengarse de los 11 sabios y del “bien”esperando la vuelta de su amo.

Muchos siglos despues, la oportunidad tan buscada por los “huecos” había llegado, el momento que habían esperado tanto tiempo había llegado. El eclipse de las “12 lunas” estaba por llegar, y, durante este fenómeno, debido a la cantidad de esencias mágicas generadas, sería el momento en el que los “huecos” deberían intentar despertar a su amo, ya que de por sí se encontraban en un estado débil perpetuo.

Debido a todo el tiempo el cual habían cohexisido con el “mal” sellado, tanto los 11 sabios como el lado de “luz” se despreocuparon, despreocupación que los “huecos” aprovecharon para reunirse co su amo y recitar el ritual “teridiano” el cual rompería la maldición de su amo, este, era un ritual que le fue otorgado debido a su anterior estado como parte del primer sabio y este, como era desconfiado por naturaleza, compartió con todos sus seres malignos por si llegaba el día en el que necesitaría concentrar parte de la magia del eclipse, en este caso, para romper el sello.

En el eclipse de las 11 lunas, la cuál ocurre 1 vez cada X sigllos, para poder despertar a su amo de su profundo sueño inducido deberían unirse en los primeros instantes del eclipse y recitar el ritual “teridiano” el cual rompería la maldicion ssobre su amo.

El juego empieza en el mometo el cual el “MAL” es sacado de su “eterna condena” y la misión será la de ir restaurando el equilibrio entre el bien y el mal (ya que comienza en un estado de debilidad tras 100 años sellado) y debido a sus actos, vengarse de aquellos los cuales le sellaron. Para conseguir dicho fin, el usuario deberá controlarle y avanzar en la aventura, comenzando con nuestro compañero principal –“NoName”-, este, nos ayudará a lo largo de toda la historia,

consiguiendo ir avanzando, completando misiones, templos, consiguiendo cofres, desbloqueando armas, compañeros (se espera poder conseguir de compañeros a los 7 pecados capitales).

6.6 Estética del juego

En esta sección se describirán distintos aspectos del juego relacionados con su estética.

El apartado gráfico se basa en un estilo “oscuro”, es decir, predominaran los colores oscuros debido a la tematica del videojuego.

Los niveles del modo desafio se ambientan en el “templo-oscuro”, templo el cual se implementará en el modo historia, este es un templo a nivel de las nubes y debido a su altitud se encuentra congelado

Los compañeros de nuestro personaje tendran formas diversas debido a que estan relacionados con los 7 pecados capitales.

Los enemigos tendran diversas formas, desde soldados del “bien” hasta monstruos en ambos bandos (enemigos/amigos)

Personajes

General (Personaje principal, enemigos, aliados, neutro)

General- Jugador/Enemigos/Aliados/Neutro					
Jugador Principal					
Enemigos	Soldados: 				
	Monstruos/Mobs: 				
	Jefes: 				
Aliados	Compañeros: <table border="1" data-bbox="755 1438 1291 1564"> <tr> <td>Compañero Principal: 7 pecados capitales</td> <td>Futura Implementación</td> </tr> <tr> <td></td> <td></td> </tr> </table>	Compañero Principal: 7 pecados capitales	Futura Implementación		
	Compañero Principal: 7 pecados capitales	Futura Implementación			
					
Monturas: 					
Neutro	NPC 's: 				
	Fauna: 				
	Flora: 				



Nuestro personaje se tratará de un “ser” oscuro debido a que representa el “mal”, el lado oscuro de la humanidad, por ello su diseño esta caracterizado por un atuendo oscuro que le tapa por completo, evitando mostrar hasta su rostro... a ello le acompaña un bastón magico el cual le ayudara en su aventura.

Para realizar sus animaciones, se han creado distintos diseños sobre las distintas posiciones a las que podrá optar nuestro personaje:

Animación general movimiento:



Animación Muerte:



Animación ataque:

Se ha creado directamente en unity con el posicionamiento relativo del arma sobre el personaje.



En este modo de juego, vemos como nuestro personaje aparece directamente a lomos de una de sus posibles monturas.

Su misión sera la de recuperar todos los cristales de sellado por el mapa que le sean posibles para así, desbloquear el futuro modo campaña/historia, batir nuestro propio record y en un futuro, desbloquear distintas monturas.

Animación movimiento +salto+idle:



Personajes Enemigos /Aliados/Neutros

Modo Plataforma

Este modo en principio sólo contará con la dificultad de intentar superar nuestro record a través de la obtención de la mayor puntuación posible, por lo que no contará con enemigos ni aliados.

En un futuro se implementarían NPC's a lo largo del camino, que nos irían animando a lo largo del viaje.

Modo Desafío & Modo Campaña

Tanto en este modo de juego como en el futuro modo campaña, encontraremos diversos enemigos, cada uno con características propias ya sea tamaño, habilidades, vida...

Los enemigos actualmente sólo poseen movimientos aleatorios, pero en un futuro se implementarán:

- IA capaz de seguir a nuestro personaje.
- Enemigos capaces de disparar a larga distancia.
- Objetos decorativos que hagan daño a nuestro personaje. (Ejemplo: Cuchillas)

Actualmente se encuentra con funcionalidad "completa":

- Tompli/totoro

Se trata del único enemigo actualmente, su movimiento es Random y su animación es la siguiente



Objetos

Modo Plataforma



Los cristales serán los objetos que nuestro personaje deberá ir cogiendo a lo largo del nivel infinito, estos cristales, en este modo de juego, serán para intentar salvar el sellado que recae en él sobre el modo campaña, desbloqueando el modo campaña al llegar a un número determinado.

Tras esto la misión será la de conseguir el máximo número de cristales posibles, luchando contra nosotros mismos para batir nuestro propio record.

Modo Desafío

En este modo de juego se van a encontrar diversos tipos de objetos, los cuales podemos categorizar en:

Misiones:

Para determinadas misiones se nos podrá pedir que recojamos diversos objetos a lo largo del mapa.



Ejemplo: Cristal → iniciará el primer nivel del modo desafío.

Armas: (No implementado)

A lo largo de los niveles podremos ir recogiendo armas auxiliares que nos podrán otorgar diversas habilidades en el ataque de nuestro personaje.

Inventario: (no implementado)

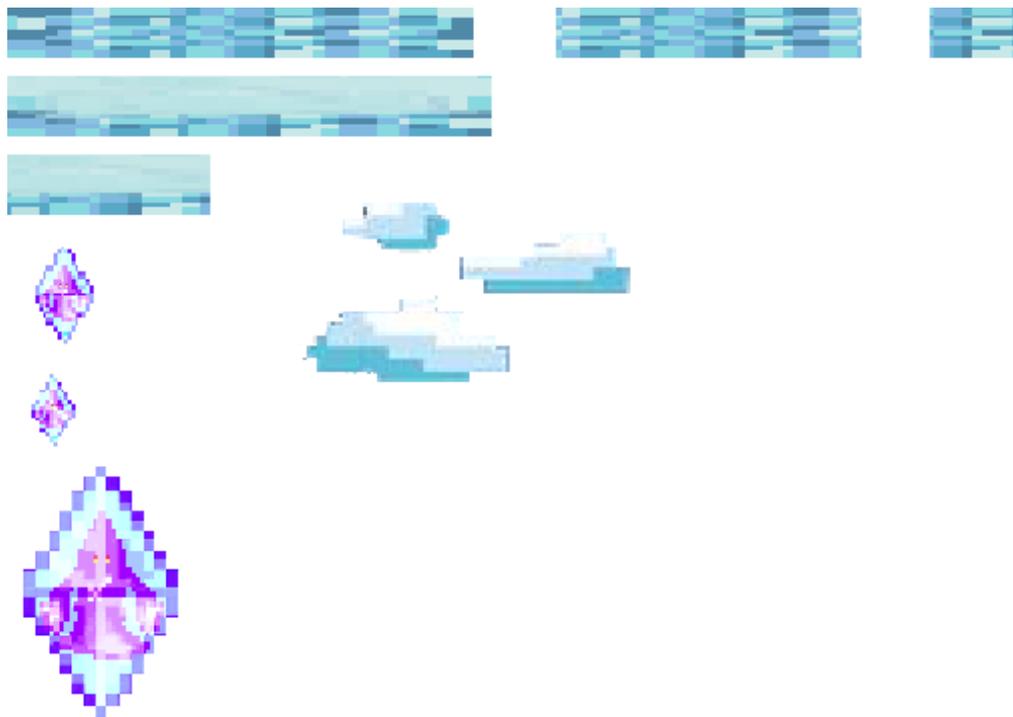
Se caracterizara por ser un inventario simple, con pergaminos de ritual (permitir´sn otorgar vida a nuestro personaje, “mana” (el cual sera necesario para el uso de determinadas armas))...

Modo Campaña

El modo capaña tiene cuyo objetivo un mundo extenso en el que podremos encontrar todo tipo de objetos, desde armas, hasta pociones, libros de magia, y todo tipo de objetos interactivos a lo largo del mundo.

Entorno

Modo Plataforma



El nivel infinito estará compuesto principalmente por 3 generadores de suelo, los cuales utilizarán los bloques de suelo para ir generando el nivel, y 2 generadores de cristales (no implementado el uso de un tercer que vaya variando su posición e introduzca en el mapa cristales con una mayor puntuación para el jugador).

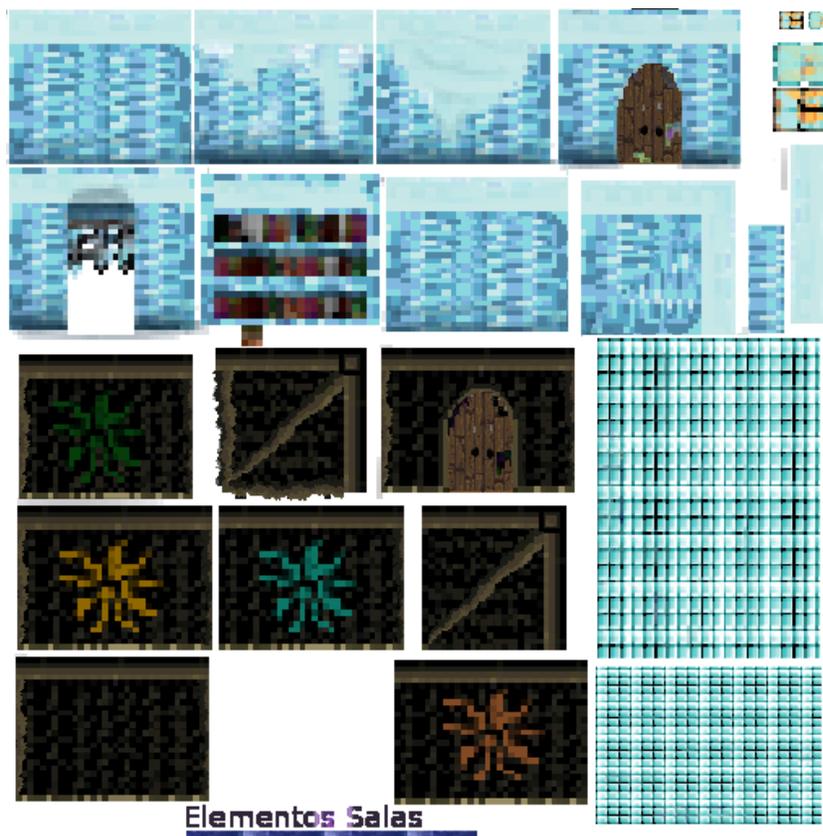
En un futuro, los generadores de plataformas tendrán, además, nuevos elementos, como nubes las cuales caerán al vacío cuando nuestro personaje se coloque encima de ellas.

Modo Desafío

Podemos clasificar los elementos decorativos del entorno en varios sub-elementos decorativos.

- Elementos de entorno de sala: Se caracterizarán por componer las salas por las que nuestro personaje irá moviéndose
- Elementos de entorno decorativo: se caracterizarán por componer la decoración dentro de cada sala.
- Elementos decorativos “recompensa”: Se caracterizarán por encontrarse a lo largo del mundo, y nos ayudarán en nuestro largo camino, ya que dentro de ellos podremos obtener diversos objetos, pero cuidado...no vaya a ser trampa!.

Elementos de entorno de sala:



Algunas de los objetos dedicados al entorno de las salas, tanto paredes como suelos, serán algunos de los mostrados anteriormente.

Elementos de entorno decorativos:

Ademas, también se caracterizará por tener múltiples items decorativos, de los cuales algunos podran llegar a iniciar misiones o ayudar al jugador a pasar el nivel otorgandole pistas, pistas dadas directa o indirectamente al jugador...



Elementos Decorativos

Elementos de entorno decorativos “Recompensas”:

La última clase de items decorativos serán los cofres, estos, estaran distribuidos a lo largo de los niveles y en ellos podremos encontrar desde bonificaciones para el nivel hasta armas.

Estos seran algunos de los diseños, pero cabe destacar que a lo largo del juego encontraremos un gran número de diversos tipos de cofres.

Cofre oculto especial: Compañeros especiales (posible desbloqueo mediante el uso de los tres modos de juego)

Cofre transparente: Elementos de mejora de compañeros

Cofre armas/amadura: Armas/armaduras de combate

Cofre ítems ayuda: Podremos encontrar objetos mundanos como pociones, revivires...

Cofres compañero: En ellos podremos encontrar a lo largo de la historia compañeros que nos ayudaran a lo largo del camino, otros compañeros simplemente se podrán conseguir según vaya avanzando la historia.



Chests

Interfaz de Usuario

La siguiente tabla muestra una platilla de cómo van a ir organizados/definidos las especificaciones de la interfaz de usuario.

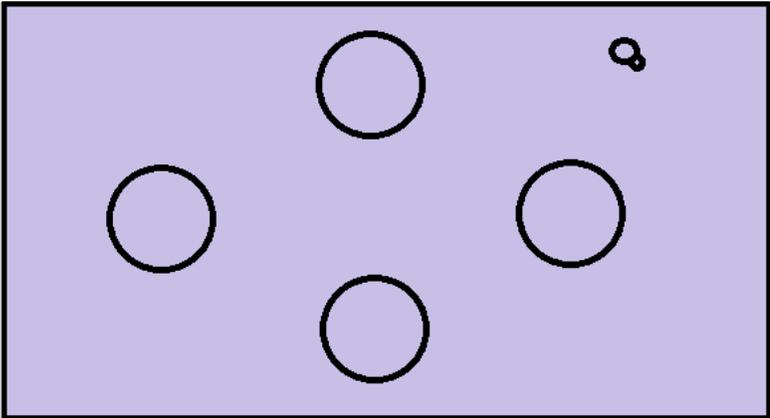
ID y Nombre	IU-XY
Descripción	Breve descripción
Disparador	Acción/es que activan la interfaz
Boceto	Boceto que representa el prototipo de la interfaz
Eventos	Eventos que pueden generar
Otra información	Otra información de interés

Tabla 95. Ejemplo tabla IU

Comunes

S

Se recogerán aquellas interfaces de usuario comunes para todos los modos de juego

ID y Nombre	IU-01 Pantalla de selección de juego (HomeScene)
Descripción	Pantalla principal del juego la cuál nos va a mostrar diferentes opciones como: Elección entre los 3 modos de juegos disponibles, salir de la aplicación o entrar en el menú de configuración.
Disparador	<ol style="list-style-type: none"> 1. El jugador ejecuta el juego en su dispositivo móvil y espera al video inicial. 2. El jugador pulsa el botón “Volver Menú Principal” en el menú GameOver del modo plataforma. 3. El jugador pulsa el botón “Volver Menú Principal” en el menú GameOver del modo desafío. 4. El jugador pulsa el botón “Volver Menú Principal” en el menú de Pausa del modo desafío.
Boceto	
Eventos	ConfigurationMenuManager.HiddenMainMenu(UI Buttons) ConfigurationMenuManager.HiddenPauseMenu(ConfigurationPanel)
Otra información	Se tratará de una interfaz sencilla que intentará, al ser botones pura-

mente visuales, el ser más agradables para el usuario/jugador,

Tabla 96.IU-01 HomeScene

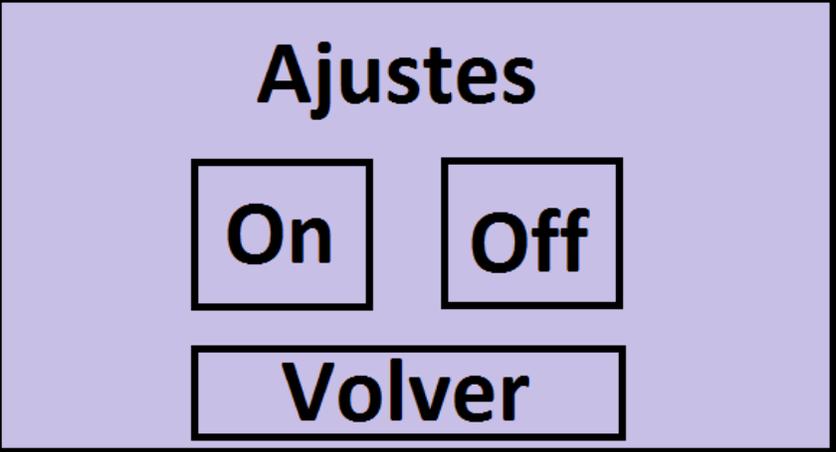
ID y Nombre	IU-02 Configuración
Descripción	Menú encargado de la gestión de los ajustes/configuración del juego, será el encargado de permitir la gestión de la música del juego.
Disparador	El jugador pulsa el botón ajustes
Boceto	
Eventos	GameSoundState.OnMusic(). GameSoundState.OffMusic().
Otra información	En versiones posteriores, se implemetarán también los efectos del juego.

Tabla 97.IU-02 ConfigurationPanel

ID y Nombre	IU-03 MainScene
Descripción	Pantalla principal del modo de juego Plataformas, esta nos mostrará una interfaz sencilla en la que podremos acceder al juego (GameScene) de forma sencilla, o bien, volver al –HomeScene–
Disparador	El jugador pulsa el botón “Play”
Boceto	
Eventos	<pre>GameManager.RestartPlatform(); GameManager.RetryHome();</pre>
Otra información	El fondo final se decidió incluir a este boceto simple montañas y nubes para dar sensación de altura, ya que las plataformas de las que constará están “congeladas”.

Tabla 98.IU-03 MainScene

ID y Nombre	IU-04 HUD Desafío - GameScene
Descripción	Pantalla de juego del modo Plataformas, se caracterizó por encontrar el HUD de este modo de juego, el cual se nos mostrara con una pequeña miniatura de nuestro personaje, y seguido, la puntuación actual.
Disparador	<ol style="list-style-type: none"> 1. El jugador pulsa el botón "PLAY" de la MainScene 2. Pulsa el botón "Nueva partida" en el panel GameOver.
Boceto	
Eventos	Input.GetMouseButtonDown(0) → Detectar que se toco pantalla, con lo que se movería nuestro personaje hacía la derecha,
Otra información	

Tabla 99.IU-04 HUD Desafío – GameScene

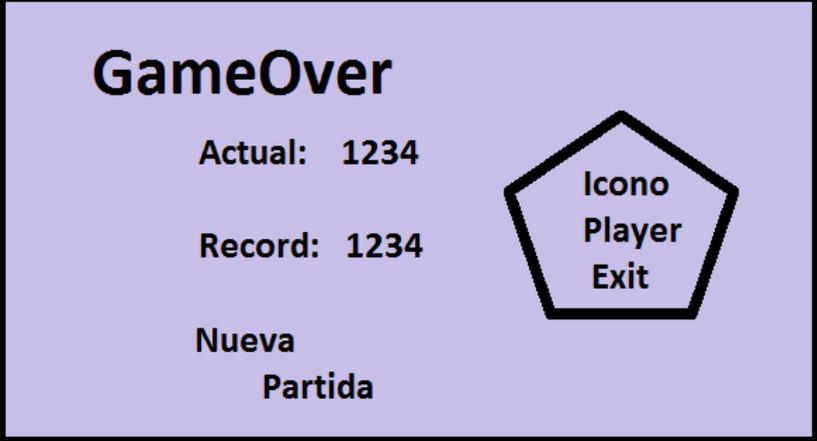
ID y Nombre	IU-05 GameOver Plataforma
Descripción	Menú GameOver, esté estará disponible una vez que nuestro personaje caiga al vacío y nos permitirá poder volver a intentar el nivel o, volver a la escena de selección de juego –HomeScene-
Disparador	El jugador se precipita al vacío.
Boceto	
Eventos	<pre>GameManager.RetryHome(); GameManager.RestartPlatform();</pre>
Otra información	El icono del jugador, será la misma utilizada para jugar con la excepción que el personaje estaría desmontado y “triste por salir del juego”

Tabla 100.IU-05 GameOver Plataforma

ID y Nombre	IU-06 SeleccionLvl
Descripción	Pantalla de selección de nivel, se caracterizó por encontrar una lista con los niveles actuales tanto desbloqueados como bloqueados
Disparador	a El jugador pulsa el “botón de juego” “Desafío” en la HomeScene.
Boceto	<p>El boceto muestra una interfaz de usuario con un fondo púrpura claro. A la izquierda, un botón cuadrado contiene el número '1'. Una flecha apunta desde este botón hacia abajo, con el texto 'Nivel' y 'Desbloqueado' debajo. A la derecha, tres botones cuadrados contienen iconos de diamantes. Una línea horizontal los agrupa, y una flecha apunta desde esta línea hacia abajo, con el texto 'Niveles' y 'Bloqueados' debajo.</p>
Eventos	
Otra información	<p>Ahora mismo solo encontraríamos un Script simple cuya función será únicamente la de cargar el primer nivel.</p> <p>Además, en un futuro se podrá llegar a esta escena desde el menú de pausa de cada nivel.</p>

Tabla 101. IU-06 SeleccionLvl

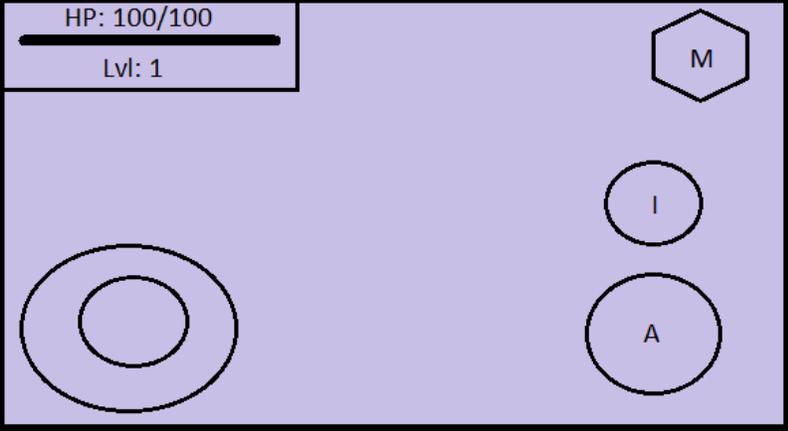
ID y Nombre	IU-07 HUD Modo Desafio
Descripción	Pantalla de juego del modo Desafio, se caracterizá por encontrarse tanto información del jugador como los botones con los que interactuaremos con el juego. <ul style="list-style-type: none"> - Joystick: Abajo Izquierda -Botón Atacar→ A -Botón Interactuar→ I -Botón Menú→ M -HUD Vida+ Nivel→ Arriba derecha
Disparador	El jugador selecciona una nivel
Boceto	
Eventos	<pre> ButtonHandler.setDownState(Attack); ButtonHandler.setDownState(Interactive); MenuPausaManager.HiddenMainMenu(PrincipalUICancas) MenuPausaManager.HiddenPauseMenu(PausePanel) CrossPlatformInput.GetAxisRaw("Horizontal"); CrossPlatformInput.GetAxisRaw("Verical"); </pre>
Otra información	Se ha intentado, acorde con el tema del tfg, ajustar los botones a las necesidades. <ul style="list-style-type: none"> -Joystick: Colores oscuros -Botones: Inspirados en las antiguas "Maquinas Arcades".

Tabla 102. IU-07 HUD Modo Desafio

ID y Nombre	IU-08 MenuPausa Desafio
Descripción	<p>Menu de PAUSA, se encontrarán aquí diversos elementos:</p> <ul style="list-style-type: none"> -Botón Atrás→ volver a juego (V) -Botón guardar →Guarda partida(No implementado) (G) -Botón guardar y salir→Guardar y salir a -HomeSceme- (G/S) -Botón Reiniciar→ Reiniciar el nivel (R) -Botón salir→ volver a -HomeScene- (S) -Botón Inventario→ Items que nos ayudarán en los desafíos (No Implementado)(I) -Botón Configuración→Abre el panel ajustes/configuración sonido. (C)
Disparador	El jugador pulsa el botón Menú
Boceto	
Eventos	<pre> MenuPausaManager.HiddenMainMenu(PrincipalUICancas) MenuPausaManager.HiddenPauseMenu(PausePanel) ConfigurationMenuManager.HiddenMainMenu(PausePanel) ConfigurationMenuManager.HiddenPauseMenu(ConfigurationPanel) GameManager.QuitGameMode(); GameManager.RestartLvl(); </pre>
Otra información	Cabe destacar que, al “llamar” al menú pausa, el juego seguirá corriendo, no se detendrá, no siendo la “pausa” clásica y añadiendo dificultad al juego.

Tabla 103. IU-08 HUD MenuPausa Desafio

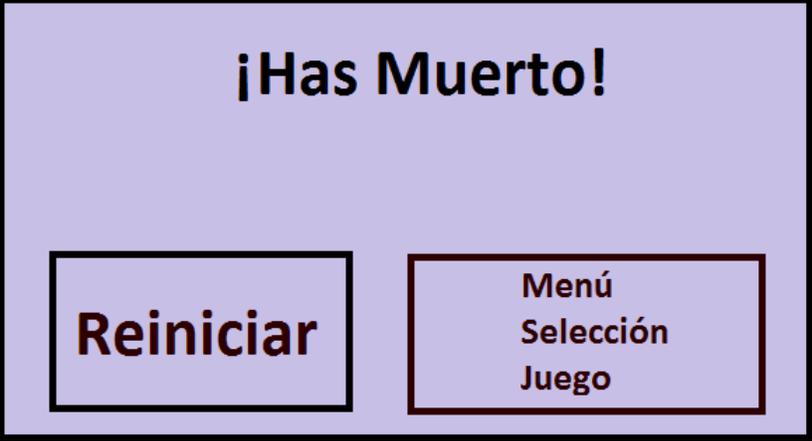
ID y Nombre	IU-09 MenuGameOver Desafio
Descripción	Menu de GameOver, nos mostrara una pantalla sencilla en la que nos permitirá actualmente el reinicio del nivel o su salida a la escena principal –HomeScene-
Disparador	La vida del jugador llega a 0 o menos.
Boceto	
Eventos	<pre>GameManager.QuitGameMode(); GameManager.RestartLvl();</pre>
Otra información	

Tabla 104. IU-09 MenuGameOver Desafio

ID y Nombre	IU-10 Ventana de diálogo
Descripción	Diálogos de texto(C) que ayudaran al jugador en los distintos niveles, proporcionándole información.
Disparador	El jugador pulsa el botón “Morado”(Interactuar) en una zona de diálogo. El jugador se acerca a una zona de diálogo con detección automática del personaje
Boceto	
Eventos	
Otra información	

Tabla 105. IU-10 Ventana Dialogo

ID y Nombre	IU-11 Pantalla Mundo General
Descripción	HUD principal, está contiene todos los elementos gráficos necesarios para interactuar con el juego.
Disparador	El jugador entra en modo de juego campaña.
Boceto	
Eventos	
Otra información	No implementado.

Tabla 106. IU-11 Pantalla Mundo General

ID y Nombre	IU-12 Pantalla batalla
Descripción	Menú HUD, este contiene todos los elementos gráficos necesarios para interactuar en las batallas del juego.
Disparador	El jugador entra en una batalla desde el “mundo general”
Boceto	
Eventos	
Otra información	No implementado.

Tabla 107. IU-12 Pantalla batalla

ID y Nombre	IU-13 Menú Equipo I
Descripción	HUD Equipo, desde aquí podremos acceder a todas las estadísticas de nuestro personaje o compañeros y además, otras funcionalidades como inventario o guardar partida
Disparador	El jugador pulsa el botón “Menu Equipo”
Boceto	
Eventos	Eventos que pueden generar
Otra información	No implementado.

Tabla 108. IU-13 Menú Equipo

ID y Nombre	IU-13 Menu equipo II
Descripción	HUD Equipo, desde aquí podremos acceder a las configuraciones de nuestro personaje y sus compañeros.
Disparador	El jugador pulsa el botón “Menu Equipo”
Boceto	
Eventos	Eventos que pueden generar
Otra información	No implementado.

Tabla 109. IU-13 Menu Equipo II

Capítulo 7

Implementación

En este capítulo se explica cómo se ha desarrollado e implementado la estructura y el diseño especificados.

7.1 Organización

The Dark World - The Age Of Darkness : proyecto Unity3D.

- **Assets**: paquetes de recursos.
 - **The Age Of Darkness**: Se trata de un asset en el cual se encuentran contenidos todos los recursos utilizados en el juego.
 - **animations**: Contiene todas las animaciones empleadas en construcción del videojuego. Esta, a su vez, estará organizada en subcarpetas para que su contenido este lo mas “limpio” posible.
 - HomeScene: Se encontrarán las animaciones necesarias para desplegar, o no, el menú de configuración
 - I. *ConfiguraciónPanel*: Animación cuya función es la de mostrar/ocultar el panel configuración.
 - II. *MainButtons*: Animación cuya función es la de mostrar/ocultar la interfaz principal con los botones de selección.
 - GameModes: Se encontrarán las animaciones de los distintos modos de juego.
 - ChallengeGameMode: Carpeta contenedora de las animaciones del modo de juego -Desafío-
 - I. Enemies/totoro –**Totoro01**- Animación de los enemigos “Tomplis”, la diferencia de nombre se debe a que aún no se ha decidido su nombre final.
 - II. MenuPausa –**MainPanelCanvas**- Animación cuya función es la de mostrar/ocultar la interfaz principal (joystick, slider HP...)
 - III. MenuPausa –**PausePanelCanvas**- Animación cuya función es la de mostrar/ocultar el menú de pausa
 - IV. TempleAnimationPJ/PlayerAttack: Animaciones de ataque del personaje principal.

- V. TempleAnimationPJ/PlayerMovement: Animaciones de movimiento del personaje principal
- **PlatformGameMode**: Carpeta contenedora de las animaciones del modo de juego Plataforma-
 - I. MainScreen –**AnimationPlayButton**- Animación que permitirá mayor dinamismo en el botón “PLAY” para comenzar el juego.
 - II. GameScreen/MainPlayer –**Idle**- Animación del personaje principal, se mostrará cuando el personaje este parado.
 - III. GameScreen/MainPlayer –**Jump**- Animación del personaje principal, se mostrará cuando salte.
 - IV. GameScreen/MainPlayer –**Run**- Animación del personaje principal, se mostrará cuando el personaje comience su movimiento horizontal.
 - V. GameScreen/MenuGameOver –**GameOver**- Animación que se mostrará cuando el jugador caiga al vacío.
- **fonts**: Fuentes de texto usadas en el juego.
- **materials**:
- **materials-physics**: materiales de las físicas de los objetos.
 - slippery: material “resbaladizo”, se encargará que nuestro personaje en modo –Plataformas- no se quede “pegado” a las plataformas.
- **Prefabs**: Se encuentran las plantillas de los objetos, es decir, objetos los cuales se quiere que mantengan propiedades específicas para su posterior utilización.
 - **General**: Carpeta contenedora de aquellos prefabs que se utilizarán a lo largo de todo el proyecto.
 - **GameModes**: carpeta contenedora de aquellos prefabs específicos a cada modo de juego.
 - **ChallengeGameMode**
 - I. enemies: Prefabs enemigos y componentes a utilizar con ellos.
 - II. joystick: Prefab joystick
 - III. quest: Prefabs Canvas Quests/misiones
 - IV. rooms: Prefabs de las distintas salas que conforman los niveles.
 - V. componentZone: Prefabs de los distintos componentes más utilizados en la escena (Ejemplo: Main Camera)
 - VI. player+weapon: Prefabs del personaje principal, así como de su arma y los elementos que influyen tanto en la estética de la colisión contra un enemigo, como la función principal de

realizar daño.

➤ **PlatformGameMode**

- I. Platform: Prefabs de las plataformas e items sobre las que nuestro personaje ira avanzando, asi como, de los generadores de estos.
- II. Others: Otros prefabs, ejemplo- BackGround.
- III. Player: Prefab del personaje principal

➤ **Others:** Otros prefabs

- **Scenes:** Se encuentran las escenas/pantallas del juego.
- **Scripts:** Se encuentra los distintos archivos de código que dotan de funcionalidad al juego, el lenguaje de estos es C#.
 - **General:** Se encuentran los Scripts que serán utilizados indistintamente del modo de juego en el que nos encontremos, también se encontraran todos los scripts de los menús.
 - **IntroScene:** Se encuentra el script encargado de mostrar un video al inicio del juego. (Actualmente esta inhabilitado)
 - **HomeScene:** Se encuentra el script encargado de mostrar el panel –Desafio-.
 - **GameModes:** Se encuentran los scripts de los distintos modos de juego
- **Sounds:** Se encuentran los distintos tipos de sonidos en el juego, tanto música del juego como efectos de sonido).
- **Sprites:** Se encuentran las imágenes en 2D utilizadas a lo largo del proyecto.
 - **FutureSprites:** Algunos de los sprites usados en un futuro.
 - **GameIcon:** Icono de juego.
 - **GameModes:** Imágenes a utilizar en los distintos modos de juego.
 - **HomeScene:** Imágenes a utilizar en la escena de selección –HomeScene-
 - **JoystickAndButtons:** Imágenes a utilizar en la interfaz principal del modo –Desafio-.
 - **Terrains:** Imágenes de terreno generales
- **Standard Assets:** Asset gratuito de la tienda de unity en lo relativo a la gestión de los controles multiplataforma.
- **StreamingAsset:** Se encuentra el video inicial del juego.

7.2 Pantallas de juego

En esta sección se describe brevemente las distintas pantallas que encontraremos a lo largo del juego.

Para ello, antes de nada, se describirán componentes comunes entre los distintos tipos de escenas.

- Canvas: Sistema de GUI, encargado de manejar la interfaz de usuario.
 - Rect Transform: Este elemento, cabe destacar, que como se comentó anteriormente, éste no tendrá la característica Transform, esto se debe a que su tamaño no depende de si mismo, sino del próximo componente.
 - Canvas: Especifica de que va a depender el canvas, en mi caso será de la cámara principal
 - Canvas Scaler: Dota de escalabilidad al sistema, ya que, en este componente se podrá seleccionar, en este caso, que se escale en proporción a la tamaño del dispositivo en el que se este ejecutando (Atributo- Ui Scale Modo- “ with Screen Size “)
 - Graphic Raycaster: Permite detectar eventos en la pantalla.
- En versiones anteriores de unity, este, no poseía de dicho sistema de UI por lo que el adaptar la UI era más tedioso (podemos ver que debido a la realización del backGround en el modo plataformas por medio de elementos ajenos al canvas se utilizara un script que lo ajuste)
- EventSystem: Objeto que controla los eventos del sistema.
- GameSoundState: Controlador genérico el cual es usada por toda la aplicación, es el encargado de controlar la música de fondo.
- GameManager: Controlador genérico que actualmente controla muchas de las funciones del juego, pero, principalmente ahora, la de cargar/guardar la puntuación del modo Plataforma, el objetivo sería que cuando se implemente el tercer modo de juego, los tres modos se centralicen en este manager.

Después de la entrega y exposición del proyecto uno de los objetivos será centralizar mediante un “singleton” los tres modos de juego.

Esto, se trata de un patrón el cuál tiene como objetivo el asegurarnos que solo exista una instancia de la clase proporcionando acceso fácil a ella de forma global.

Actualmente para paliar este problema nos encontramos en determinados elementos, dentro de su script, la secuencia: DontDestroyOnLoad, cuya función es la de prevenir que un elemento sea destruido, aun cambiando de escena.

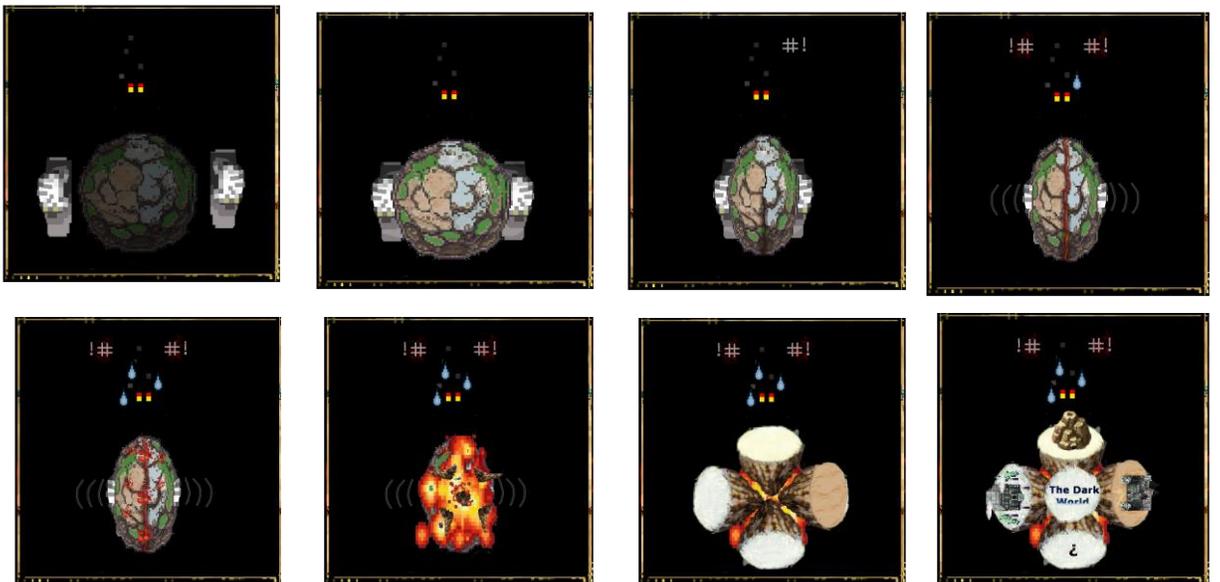
Escena: IntroScene

Se trata de la escena inicial del juego, esta, no es una escena como tal, ya que el usuario no podrá interactuar con ella.

Se encargará de mostrar al usuario un video introductorio al juego, tras éste se mostrará la pantalla de selección de juego –HomeScene-

Los elementos que podemos encontrar en esta escena son:

- Main Camera: Contendrá el script “x” el cuál es el encargado de iniciar el video al entrar en la aplicación.
- Actualmente se encuentra inhabilitada debido a que actualmente solo deja reproducir videos después de pasar al plan “PRO” de Unity, se intentará en un futuro solventar este problema creando la animación de video directamente sobre Unity.



Escena: HomeScene

Se trata de la escena principal del juego, ya que, desde aquí podemos acceder a todas las escenas del juego.

Se podría decir, que esta escena es la escena de “selección multijuego”, ya que en ella se encontrarán los distintos botones para acceder a los distintos modos de juego, así como para poder salir de la aplicación y configurar la música del juego.

- Main Camera: Cámara principal de la escena, nos mostrara la interfaz de selección de juego.
 - showPanel: script que permite mostrar o no un panel al seleccionar el botón de selección “Desafío”, este panel nos dejará entrar al juego y en un futuro ver las clasificaciones de tiempo de los distintos niveles.

- Canvas: Se encontrarán tanto la interfaz principal (Botones selección juego/salir/configuración), el panel desafio anteriormente nombrado, y la interfaz del menú de configuración.

- Managers: Se encontrarán los controladores tanto de la música del juego como del menú de selección y configuración.
 - ConfigurationMenuManager: Encargado de mostrar o no el menú de configuración dependiendo si el usuario presiona o no el botón táctil –Configuración-
 - GameSoundState: Ya comentado anteriormente.
 - MainMenuManager: Encargado de dotar de funcionalidad a los botones de selección principales.
 - GameManager: Primera especie de singleton encargado de gestionar el modo Plataforma, teniendo distintas funciones para la gestión de ese modo de juego (botones...) como guardando la puntuación “Record” obtenida en dicho modo de juego.
 - EventSystem: Ya comentado anteriormente.

- Resultado:

- Interfaz Principal
Se puede reflejar, debido a la oscuridad de dicha escena, que posee un toque “oscuro”



- Interfaz menú configuración
Interfaz sencilla que nos permite encender o apagar la música, se espera implementar con el tiempo un menú mas completo.



Escena: Modo_Plataforma- MainScene

Se trata de la escena introductoria a dicho modo de juego, en ella nos encontraremos un botón “animado” el cual permitirá interactuar con él para el acceso al juego.

- Main Camera: Cámara principal de la escena, nos permitirá visualizar la interfaz de dicha escena.
 - Background: Contiene dos elementos de Background, los cuales contienen un script llamado “Scroll” cuya función es la de realizar Parallax Scroll, es decir, añadiendo una velocidad a cada uno de los fondos se consigue sensación de profundidad
 - Cabe destacar, que, para practicar distintos métodos, este background no utiliza la GUI proporcionada por Unity y la cuál

se ajusta automáticamente a la pantalla, por lo tanto se ha tenido que implementar un scrip “Adaptative Resolution” que será la encargada de realizarlo.

El objetivo de realizarlo de dicho manera es para adquirir distintos conocimientos.

- Canvas: Se encontrara el único botón de la escena y el cuál es el encargado de llevar al usuario al juego, esté, consigue su funcionalidad mediante la utilización del “GameManager” controlador del juego.
- EventSystem: Ya comentado anteriormente.
- GameState: Ya comentado anteriormente.
- GameManager: Controlador cuya función tendrá la de guardar el progreso del juego, en este modo de juego, así como el de proporcionar funcionalidad a determinados botones.
- Resultado:



Escena: Modo_Plataforma- GameScene

Se trata de la escena de juego de este modo.

- Main Camera: Cámara principal de la escena, nos permitirá visualizar la interfaz de dicha escena, y, además, posee los siguientes elementos:
 - Generadores: Se encuentran contenidos los generadores de plataformas y cristales, los cuales se pondrán en funcionamiento en el instante en el que el usuario empiece su movimiento.
Para ello poseen el script: “Generator” encargado de generar aleatoriamente objetos en determinadas posiciones.
 - destructores: Se encuentran los dos destructores de nivel.
 - Destructor Izquierdo: Su función es la de destruir todas aquellas plataformas que deje atrás el personaje, para así, evitar sobrecargas de objetos.
 - Destructor Abajo: Su función es la de detectar que nuestro personaje se ha precipitado al vacío activando el menú GameOverPara ellos, ambos destructores poseerán el script “Destructor”
 - Background: Fondo de pantalla con función “Parallax Scroll”.
- InitialStructures: Objeto en donde se encuentran englobadas las plataforma e ítems iniciales del nivel.
- Canvas: Se encontrará el HUD del nivel (Puntuación) y el menú GameOver.
- Player: Personaje principal, se hablará de él en la siguiente sección.
- EventSystem: Ya comentado anteriormente.
- GameSoundState: Ya comentado anteriormente.
- GameManager: Controlador cuya función tendrá la de guardar el

progreso del juego, en este modo de juego, así como el de proporcionar funcionalidad a determinados botones.

- Resultado:



Nota: Cabe destacar que se ha hecho uso de un script llamado: NotificationCenter, el cual nos permite una comunicación fácil, este, no se le ha de asignar a ningún GameObject y, por defecto, al iniciar el juego se creara un objeto de él.

Escena: Modo_Desafío-SelectionLevel

Se trata de la escena de juego del modo –Desafío–.

- Main Camera: Cámara principal de la escena, nos permitirá visualizar la interfaz de dicha escena.
- Canvas: Se encuentra la interfaz de la escena (Background+niveles)
- MenuManager: Actualmente es un objeto controlador muy básico, ya que sólo proporciona funcionalidad al botón “nivell”, ir al nivell modo desafio.
- EventSystem: Ya comentado anteriormente.
- GameSoundState: Ya comentado anteriormente.

- Resultado:



Escena: Modo_Desafío-Zone1MainRoom

Se trata de la escena de juego, en este caso, y como forma de ejemplo, se llevará a cabo la descripción del nivel 1.

- Main Camera: Cámara principal de la escena, nos permitirá visualizar la interfaz de dicha escena, además, posee los siguientes elementos:
 - Camera Controller: Se encargara de, indicándole a quién y qué velocidad usar, seguir el movimiento de nuestro personaje.
- Canvas (UI General): En ella se encuentran las distintas interfaces que serán utilizadas a lo largo de este modo de juego
 - DialogCanvas: Interfaz para los cuadros de dialogo, en caso que sean necesarios.
 - PauseMenuCanvas: Interfaz para el menú pausa, además, desde dicho menú podrá accederse al *menú configuración*. Este menú se podrá mostrar como una animación activada desde el botón “MENU” en la interfaz principal.
 - Encontraremos implementados actualmente los botones de reinicio de nivel, volver al juego y salir a -HomeScene-.
 - PrincipalUICanvas: Interfaz principal de este modo de juego, en ella se encuentran:
 - BotónMenuPausa: Es el encargado junto con el con-

trolador que veremos después (MenuPausaManager) de activar/desactivar el menú de pausa a través de una animación que ocultará un menú y mostrará otro.

- DialogueManager: Encargado de gestionar los diálogos. Este, posee el script “DialogueManager”
- MobileSingleStickControl: Joystick y botones de juego.
 - Joystick: Se encuentra tanto el joystick como el “halo” de fondo, el cual completa el diseño del joystick.
 - Botones:
 - Atacar: “Botón rojo”, permite que nuestro personaje realice la animación de atacar. Para esto necesita el script Button Handler, el cual detecta que el botón ha sido pulsado, genera un *event Trigger* que llama a funciones del script Button Handler, para realizar la acción pulsada.
 - Interactuar: “Botón morado”, su funcionamiento sería igual que el botón de atacar pero este, servirá para interactuar con los elementos del mapa y, para pasar los diálogos.
- ConfigurationPanel: Interfaz del menú de configuración del sonido.
- PlayerStatisticsCanvas: Nos mostrará dos tipos de estadísticas:
 - HP: Mediante un slider + Texto, se nos mostrará la vida actual de nuestro personaje.
 - Level: Nos mostrara el nivel del personaje, este podrá ser necesario en los niveles que consistan en modos de aguante de enemigos, siendo un indicador para pasar de nivel.

Para ello cuenta con dos scripts: UIManager (Encargado de actualizar tanto el HP como el nivel de nuestro personaje) y PlayerStats (Se encontrará aquí las especificaciones para poder subir de nivel, encontrándose ahí la experiencia necesaria para ello).

- GameOverPanel: Interfaz del menú de GameOver, esta nos será mostrada después de que nuestro personaje muera en manos de nuestros enemigos un panel con las opciones de “Reiniciar Nivel” o “Salir a HomeScene”.
- Salas: Se encontrarán las salas de las que estará compuesta cada “zona”, además, se encontrarán dos objetos (StartPoint-El personaje reaparecerá en dicho punto) (PointsChangeZone-Entrada a otra zona)
- People: Se encontrarán tanto los enemigos como los NPC’s de la zona. Para esta versión del proyecto consta de varios enemigos (Totoro-tompli) éstos son el mismo, pero debido a la incapacidad para tomar una decisión con su nombre le podemos encontrar de ambas maneras, en un futuro no será así.
 - Enemigos:
 - Totoro/Tompli:** Constará de:
 - HurtPlayer: Encargado de realizar un daño de 20 de vida a nuestro jugador y, mostrar este daño mediante una indicación numérica.
 - Enemy Health Manager: Encargado de controlar la vida del enemigo, así como la experiencia proporcionada al morir y, por si fuese un enemigo de misión, el tipo de enemigo.
 - Tompli Control: Controlador del movimiento de nuestro enemigo, cabe destacar que el movimiento de este, en esta primera versión es Random.
 - NPC’s:
 - Buddy: Actualmente, este se encuentra rodeado de una zona de diálogo, la cual se activa cuando nuestro personaje entra en ella y, se pretende hacer que el primer nivel se inicie con una misión suya.
- Managers: Se encuentran distintos controladores de juego
 - MenuPausaManager: Encargado de, una vez presionado el botón “ButtonMenuPausa”, ocultar la interfaz de jue-

go(Joystick, botones...) y mostrar la interfaz de pausa(cabe destacar que este modo de juego tiene menú pausa pero no tiene “pausa de tiempo” es decir, los enemigos seguirán moviéndose para aumentar la dificultad, solos se quedará nuestro personaje congelado cuando interactúe con alguna zona de misión o npc)

- QuestManager: Contiene el script “QuestManager”, se encargara de gestionar las “Quests/misiones” del nivel.
A su vez, las “Quest” dentro de este controlador tendrán su propio script el cual les defina la misión en si.
 - GameManager: Actualmente Gestor botones
 - GameState: Ya comentando anteriormente.
 - ConfigurationMenuManager: Menu de configuración/ajustes del juego, se accedera desde el menú de Pausa- configuración.
- GeneralQuest: Se encuentran todas las “Quest/misiones” del nivel.
 - Para las quests que no tengan como objetivo el matar enemigos, contarán con dos puntos (inicio misión-finalización), por otro lado, para aquellas que tengan como objetivo el acabar con enemigos solo tendrán un punto de inicio de misión y el punto final será en donde se mate al último enemigo necesario.
 - MainPlayer: Personaje principal, se hablará de forma más extensa de él en la siguiente sección.
 - GameState: Ya comentado anteriormente.
 - EventSystem: Ya comentado anteriormente.

- Resultado:

PauseMenu



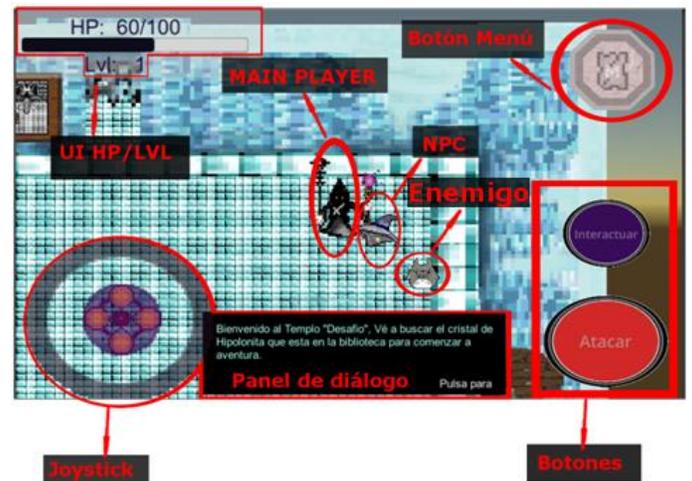
ConfiguratonMenu



GameOverMenu



General UI



7.4 Personaje

En esta sección se va a describir de manera más exhaustiva los personajes del juego.

Mini-juego Estilo Plataforma

El personaje principal de esta modalidad será "Ifrahul" el cuál ira montado en su montura para así intentar llegar lo máximo posible,

Ifrahul(Player): Los componentes básicos que esté posee serían.

- TAG-Player: Etiqueta encargada de catalogar a nuestro personaje como

Player/jugador principal.

- **SpriteRenderer:** Sprite de ifrahul montado a lomos de su montura, se trata de la imagen principal del juego pero con la característica de la montura que le ayudará a moverse entre plataformas, por defecto se trata de la primera imagen del “spritesheet” donde se encuentra.
- **PolygonCollider2D:** Componente que define el área de colisión de Ifrahul con su montura, hay otros métodos de collider pero se ha elegido este debido a que se ajusta mejor a su forma y, cabe destacar que la cabeza de la montura se encuentra fuera del collider debido a que en ocasiones se enganchara con las plataformas .
- **RigidBody2D:** Componente encargado de colocar al personaje bajo el efecto de físicas. Se a seleccionado la opción Freeze Rotation para evitar cambios de angulos en la posición de nuestro personaje .
- **AudioSource:** Se encuentra el efecto de sonido para el salto del jugador
- **Animator:** Controlador de las animaciones de nuestro personaje (Salto, Run, idle)
- **PlayerControllerPlatform:** Script encargado de gestionar el movimiento del personaje, controlando entre otras, velocidad, físicas de nuestro personaje etc.

Ademas de todos estos componentes, posee un GameObject hijo llamado “comprobar suelo” el cual será el encargado de notificar al sistema si en verdad está o no el personaje en el suelo.

Scripts:

Ahora, se va a proceder a mostrar una breve descripción para explicar las funcionalidades del personaje, para más detalle el código estará explicado en el proyecto.

PlayerControllerPlatform

- Lo primero que encontramos son las variables que se van a utilizar a lo largo del script, a grandes rasgos son componentes del personaje, variables de suelo y especificaciones de las fuerzas de movimiento y salto.

Tras esto, entramos en las funciones que podemos encontrar, no se entrará mucho en la descripción principal de las funciones debido que fueron comentadas anteriormente en el ciclo de vida de un script en unity.

- **Start():** Se encargará de obtener en las variables pjRB/pjAnimator los componentes del personaje –RigidBody2d/Animator. Con esto, podremos con-

trolar como se comportan las físicas con nuestro personaje (movimiento, salto...) y elegir las animaciones necesarias para ello.

- **FixedUpdate():** Si el personaje está corriendo, se le aplica una velocidad positiva en el eje X(En el RigidBody del personaje).

pjRb.velocity= new Vector2(m_Speed,pjRb.velocity.y) Permitirá el desplazamiento del personaje, en el eje Y dejaremos la velocidad que tenía el personaje por si acaso se encuentra saltando. Además, se iniciará la animación y se comprobará si el personaje se encuentra en el suelo, comprobando si el punto (*public Transform checkFloor*) con el radio (*private float checkRadius = 0.0625f*) colisiona con un objeto que pertenezca a la capa "Floor" y de ser así se reiniciará el doble salto.

- **Update ():**Se encargará de detectar si el usuario ha tocado la pantalla o no. (*Input.GetMouseButtonDown(0)*).

Si fuera la primera vez que se toca la pantalla, el personaje estaría en el estado "Idle" por lo que se pondría a correr, enviando una notificación de que el usuario a empezado a correr (lo que iniciaría el movimiento del fondo) *NotificationCenter.DefaultCenter().PostNotification(this, "PlayerStartToRun")*, de no ser así, se detectaría que el personaje está corriendo y lo que se debería controlar es si el personaje está en el suelo/si le quedan saltos disponibles.

En caso de encontrarse en el suelo, sobre una plataforma, el personaje podrá realizar un salto.

(pjRb.velocity = new Vector2(pjRb.velocity.x, jumpingForce)) Además, si se encuentra en el aire se le podrá adicionar un salto más, llegando a un máximo de dos. El personaje en el aire solo podrá realizar un salto extra.

Mini-juego Estilo Desafío

El personaje principal de esta modalidad será "Ifrahul" el cuál tendrá que intentar destapar los entresijos de los distintos niveles.

Ifrahul(Player): Se tratará con el "Prefab" de este personaje, y el *GameManager* será el encargado de instanciarlo en el mapa, los componentes básicos que esté posea serían.

- **TAG:** Player—Etiqueta encargada de catalogar a nuestro personaje como Player/jugador principal.
- **SpriteRenderer:** Sprite de ifrahul, es decir, se trata de la imagen principal de "Ifrahul", por defecto se tratará de la imagen Idle_Down, saliendo nuestro personaje parado mirando hacia abajo.

- **BoxCollider2D:** Componente que define el área de colisión de Ifrahul.
- **RigidBody2D:** Componente encargado de colocar al personaje bajo el efecto de físicas. Se a seleccionado la opción Freeze Rotation para evitar cambios de angulos en la posición de nuestro personaje y, debido al modo de juego, la gravedad se ha puesto a “0”.
- **Animator:** Controlador de las animaciones de nuestro personaje (Idle, Atacar, Morir, Andar)
- **Scripts: PlayerController:** Script encargado de gestionar el movimiento del personaje, controlando entre otras, velocidad, físicas de nuestro personaje etc.
- **Scripts: PlayerHealthManager:** Script encargado de gestionar la salud de nuestro personaje, el efecto cuando es golpeado por un enemigo y el activar el panel GameOver activando su componente –Animator-.

Ademas de todos estos componentes, posee un GameObject hijo llamado “Weapon” que contiene el arma con la que inicia nuestro personaje, este, tendrá los siguientes componentes:

- **BoxCollider2D:** Componente que define el área de colisión del arma.
- **Scripts: HurtsEnemy:** Script encargado de de gestionar la cantidad de daño realizada a los enemigos, el área de golpeo del arma, el indicador numérico del daño realizado y la animación cuando se produce la colisión.

Scripts:

Ahora, se va a proceder a mostrar una breve descripción para explicar las funcionalidades del personaje, para más detalle el código estará explicado en el proyecto.

PlayerController

- Lo primero que encontramos son las variables que se van a utilizar a lo largo del script, a grandes rasgos son componentes del personaje, referencia estática a este para acceder a él, especificaciones de las fuerzas ejercidas en él, acceso a movimientos “pasados” ey especificaciones sobre el estado “Atacar” de nuestro eprsonaje.
- **Start():** Se encarga de obtener los componentes del personaje y además, inicializar características sobre el movimiento del personaje.
- **Update():** Se encarga de detectar si el personaje puede moverse (Si se en-

cuentra en zona de dialogo o atacando no podrá hacerlo)y de no ser así se le “congelará”.

Si el personaje no se encuentra atacando, eso quiere decir que puede moverse. Esto se conseguirá, accediendo a “*CrossPlatformInputManager.GetAxisRaw*” el cuál nos devolverá el valor del eje virtual (rango -1...1) del joystick.

Con esto, se analizarán los cuatro casos posibles y posteriormente, para arreglar algunos bugs del movimiento del jugador relacionados con las diagonales, se llevará a cabo una “suavización” de los datos obtenidos mediante “*CrossPlatformInput*” los cuales en un principio se encuentran sin “suavizado” alguno, es decir, los datos que devuelven son -1,0,1.

Además, controlará si el usuario presiona el botón atacar y de ser así, “congelará” el movimiento del personaje, y se controlará el tiempo que esto estará así según la especificación de tiempo indicada desde Unity3D.

Tras esto enviará todos los datos al animator.

PlayerHealthManager

- Lo primero que encontramos son las variables que se van a utilizar a lo largo del script, a grandes rasgos son componentes tanto del personaje como de la animación del panel GameOver, variables de HP(Vida del personaje) y de efecto(para cuando sea golpeado).
- Start(): Se encarga de inicializar la vida actual del personaje a la máxima y obtener los distintos componentes a utilizar posteriormente.
- Update(): Se encarga de controlar si la vida del personaje llega a “0” o, si esté ha sido golpeado por un enemigo.
 - Si la vida del personaje llega a “0” o baja de “0”, entonces se “lanza” la animación de “muerte” y, cuando este acaba se mostrará el panel de GameOver.
 - Si el personaje es golpeado por un enemigo(lo veremos después de esto), se accede al “*SpriteRenderer*” de nuestro personaje y se le van modificando distintos parámetros durante un periodo corto de tiempo para acabar volviendo a su estado normal.
- HurtPlayer(): Se encarga de realizar daño al personaje, recibe un número entero cuyo valor es el daño realizado y le reduce dicha cantidad a la cantidad actual que tenga de vida nuestro personaje, además, activa el efecto de “golpeo”

- **SetMaxHealth():** Define que la cantidad de vida actual de nuestro personaje es la cantidad máxima
- **GOA():** Activa la animación del panel de GameOver, lo que hará que esté se muestre.

HutsEnemy(Weapon)

- Lo primero que encontramos son las variables que se van a utilizar a lo largo del script, a grandes rasgos: Variable numérica del daño, el gameObject asociado a las partículas generadas durante el ataque, el “punto de golpeo” del arma y el GameObject al indicador numérico
- **OnTriggerEnter2D():** Si el “hitPoint” impacta contra un enemigo, se lanzará este trigger el cual interactuará con el <EnemyHealthManager> , que gestiona la vida de nuestros enemigos con el objetivo de reducirles la vida, instancia las partículas junto en el punto de golpeo y el indicador con el daño realizado.

7.5 Enemigos

En esta sección se va a describir de manera más exhausta los enemigos del juego.

Mini-juego Estilo Desafío

En la versión actual del juego solo se puede encontrar un tipo de enemigo, y encontraremos 10 de ellos en la primera zona del mapa.

tompli(Player): Se tratará con el “Prefab” de este personaje, y el *GameManager* será el encargado de instanciarlo en el mapa, los componentes básicos que esté posea serían.

- **TAG:** Enemigo—Etiqueta encargada de catalogar a nuestro enemigo como tal.
- **SpriteRenderer:** Sprite de Totoro/tompli, por defecto se tratará de la primera imagen de su spritesheet.
- **CircleCollider2D:** Componente que define su área de colisión.
- **RigidBody2D:** Componente encargado de colocar al personaje bajo el efecto de físicas. Se ha seleccionado la opción Freeze Rotation para evitar cambios de ángulos en la posición de nuestro personaje y, debido al modo de

juego, la gravedad se ha puesto a “0”.

- **Animator:** Controlador de las animaciones de nuestro personaje (MovRandom)
- **Scripts: TompliControll:** Script encargado de gestionar el movimiento del personaje, controlando entre otras, velocidad, físicas de nuestro personaje etc. El movimiento de éste, en la versión actual se trata de un movimiento random/aleatorio el cual se ira alternando con momentos de “pausa” en los que el enemigo estará parado.
- **Scripts: EnemyHealthManager:** Script encargado de gestionar la salud de nuestro enemigo, a cantidad de experiencia que da y, por si se tratara de un enemigo de misión el nombre el cual llevara en dicha misión, por defecto se usara su nombre.
- **Scripts: HurtsPlayer:** Script encargado de de gestionar la cantidad de daño realizado a nuestro personaje por parte de los enemigos y la gestión del indicador numérico de daño.

Scripts:

Ahora, se va a proceder a mostrar una breve descripción para explicar las funcionalidades de los enemigos, para más detalle el código estará explicado en el proyecto.

TompliController

- Lo primero que encontramos son las variables que se van a utilizar a lo largo del script, a grandes rasgos son, componente RigidBody2D de nuestro personaje, y variables que afectarán su movimiento como su velocidad, tiempo requerido entre movimientos, un contador que controle ese tiempo, dirección...y demás variables que afectan a su movimiento.
- **Start():** Se encarga de obtener el componente “RigidBody2d” del enemigo y además, inicializar características con tiempos aleatorios para cada enemigo, permitiendo aún solo habiendo un tipo de enemigo añadir más aleatoriedad a sus movimientos.
- **Update():** Se encarga de dotar de movimiento al enemigo, si esté no se encuentra en movimiento se esperara el “tiempo aleatorio” necesario y se llevará a cambio un movimiento aleatorio durante un tiempo aleatorio.

Si por el contrario, esté se encontrará en movimiento, se esperará el “tiempo aleatorio” necesario y al finalizar se parará el enemigo y se le volveria a definir un “tiempo aleatorio” de espera. ($timeBetweenMoveCounter = Random.Range(timeBetweenMove * 0.75f, timeBetweenMove * 1.25f)$)

EnemyHealthManager

- Lo primero que encontramos son las variables que se van a utilizar a lo largo del script, a grandes rasgos son: Componente “QuestManager”, necesario para poder gestionar misiones dirigidas a matar determinado número de enemigos, variables de HP/vida y de experiencia.
- Start(): Se encarga de inicializar la vida actual del enemigo a la máxima y buscar el GameObject “*QuestManager*” del proyecto en unity para obtener su componente <PlayerStats> en la variable de tipo <PlayerStats> *thePlayerStats*.
- Update(): Se encarga de controlar si la vida del enemigo llega a “0” o, si esté ha sido golpeado por nuestro personaje.
 - Si la vida del enemigo baja de “0”, entonces se comprueba si es un enemigo de misión para luego “destruirle” de la escena y otorgar la experiencia especificada desde unity que reporte este enemigo.
- HurtEnemy(): Si el Enemigo es golpeado por el “ifrahul”, se le restará la cantidad especificada desde Unity3d.
- SetMaxHealth(): Define que la cantidad de vida actual de nuestro enemigo es la cantidad máxima.

HurtPlayer

- Lo primero que encontramos son las variables que se van a utilizar a lo largo del script, a grandes rasgos son: Número entero relativo al daño ejercido, y referencia al GameObject –DamageNumbers.
- OnCollisionEnter2D(): Se encarga de detectar cuando esté colisiona contra “ifrahul” y, de ser así, reducirle la vida con el daño especificado desde Unity3D, tras esto, instancia el indicador numérico de daño.

Capítulo 8

Pruebas

En este capítulo se describirán las pruebas realizadas para comprobar el correcto funcionamiento de la aplicación. Para ello, se hará uso de las *Estadísticas de unity*, estas permiten conocer como se comporta el dispositivo móvil donde se esta ejecutando la aplicación urante su ejecución.

Lo esperado para estas pruebas es obtener el mejor tiempo de respuesta posible, esto hará que mejoré la experiencia usuario-videojuego.

Las pruebas realizadas han sido llevadas a cabo con el dispositivo móvil ya comentado (BQ E4.5) con versión 4.4.2 de Android y se realizaran mediante el uso de *Unity Remote*.

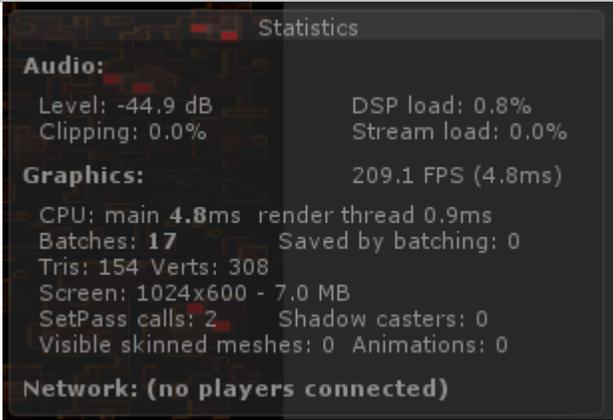
Prueba	PR-01 Abrir Menú Configuración/Ajustes
Prerrequisito	
Procedimiento	1.Pulsar el botón “Configuración” en la pantalla de inicio,
Resultado esperado	Mostrar el menú de configuración en un tiempo inferior a un segundo.
Resultado Obtenido	 <p>El menú configuración se abre en un tiempo de 0.9ms.</p>

Tabla 110. PR-01 Abrir Menú Configuración/Ajustes

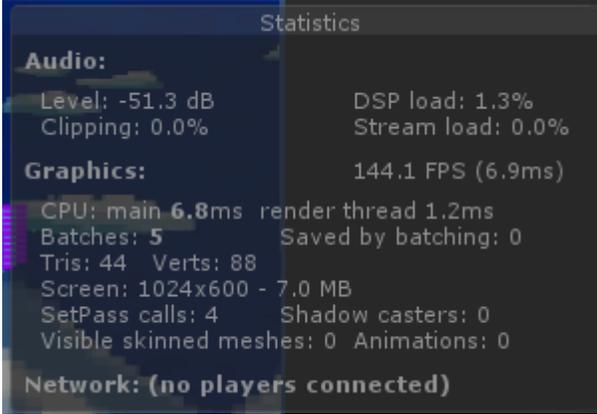
Prueba	PR-02 Entrar modo de juego Plataforma
Prerrequisito	
Procedimiento	1. Desde la escena principal –HomeScene- seleccionar el modo de juego Plataformas.
Resultado esperado	Abrir escena principal de dicho modo de juego en menos de 2 segundos
Resultado Obtenido	 <p>El nivel es cargado en un tiempo de 1.2ms</p>

Tabla 111.PR-02 Entrar modo de juego Plataforma

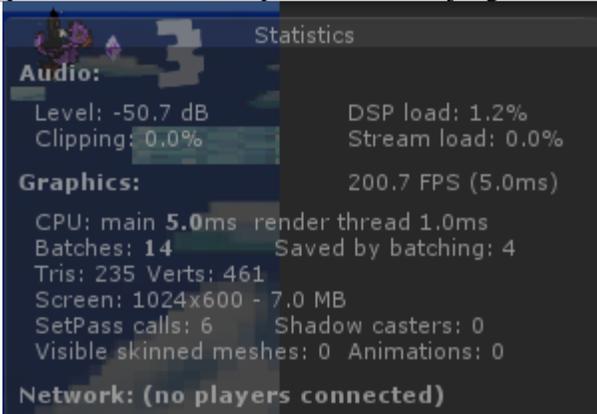
Prueba	PR-03 Salto modo Plataforma
Prerrequisito	
Procedimiento	1. Pulsar en pantalla para que el personaje salte a la que se desplaza horizontalmente.
Resultado esperado	Hacer saltar al personaje en el menor tiempo de respuesta posible, ya que de otra manera se podría hacer un juego tedioso.
Resultado Obtenido	 <p>El personaje reacciona en un tiempo de 1ms.</p>

Tabla 112.PR-03 Salto modo Plataforma

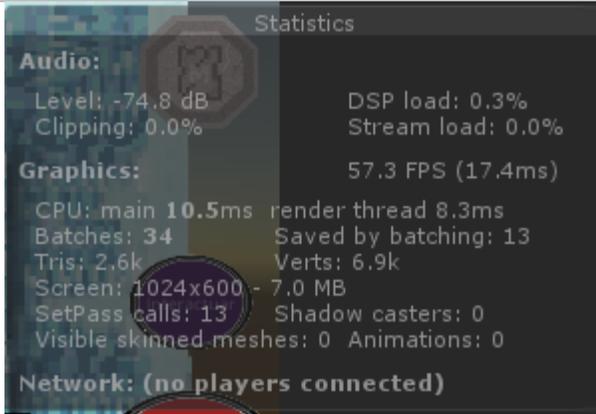
Prueba	PR-04 Cargar nivel 1 modo Desafío
Prerrequisito	
Procedimiento	1.Acceder al nivel 1 del modo desafio.
Resultado esperado	Cargar dicho nivel en menos de 2 segundos
Resultado Obtenido	 <p>The screenshot shows the following statistics:</p> <ul style="list-style-type: none"> Audio: Level: -74.8 dB, Clipping: 0.0%, DSP load: 0.3%, Stream load: 0.0% Graphics: 57.3 FPS (17.4ms) CPU: main 10.5ms, render thread 8.3ms Batches: 34, Saved by batching: 13 Tris: 2.6k, Verts: 6.9k Screen: 1024x600 - 7.0 MB SetPass calls: 13, Shadow casters: 0 Visible skinned meshes: 0, Animations: 0 Network: (no players connected)
	El nivel es cargado en un tiempo de, 8,3ms

Tabla 113.PR-04 Cargar nivel 1 modo Desafío

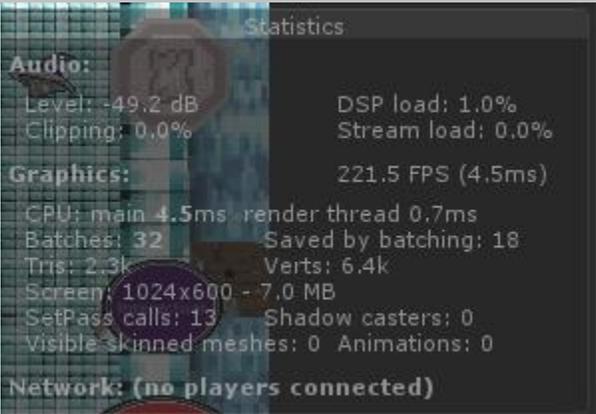
Prueba	PR-05 Movimiento + botones personaje modo Desafío
Prerrequisito	
Procedimiento	1.Pulsar el joystick y los botones de atacar e interactuar.
Resultado esperado	Mover al personaje e interactuar con sus botones en el menor tiempo de respuesta posible, ya que de otra manera, se podría hacer un juego tedioso.
Resultado Obtenido	 <p>The screenshot shows the following statistics:</p> <ul style="list-style-type: none"> Audio: Level: -49.2 dB, Clipping: 0.0%, DSP load: 1.0%, Stream load: 0.0% Graphics: 221.5 FPS (4.5ms) CPU: main 4.5ms, render thread 0.7ms Batches: 32, Saved by batching: 18 Tris: 2.3k, Verts: 6.4k Screen: 1024x600 - 7.0 MB SetPass calls: 13, Shadow casters: 0 Visible skinned meshes: 0, Animations: 0 Network: (no players connected)
	El personaje reacciona en un tiempo menor a 1ms.

Tabla 114.PR-05 Movimiento+botones personaje modo Desafío

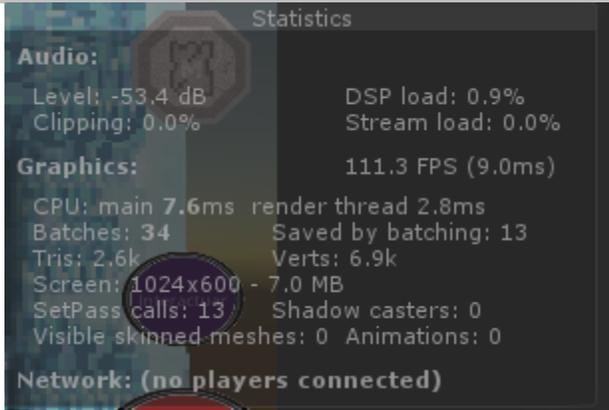
Prueba	PR-06 Reiniciar Nivel modo Desafío
Prerrequisito	
Procedimiento	1.Pulsar el botón “REINICIAR” del menú de pausa del juego.
Resultado esperado	Reiniciar nivel en un tiempo inferior a 2segundos.
Resultado Obtenido	 <p>The screenshot shows a 'Statistics' overlay from the game. Under the 'Graphics' section, it lists 'CPU: main 7.6ms render thread 2.8ms'. Other statistics include Audio (Level: -53.4 dB, Clipping: 0.0%, DSP load: 0.9%, Stream load: 0.0%), Graphics (111.3 FPS (9.0ms), Batches: 34, Saved by batching: 13, Tris: 2.6k, Verts: 6.9k, Screen: 1024x600 - 7.0 MB, SetPass calls: 13, Shadow casters: 0, Visible skinned meshes: 0, Animations: 0), and Network (no players connected).</p>
	El nivel es reiniciado en 2.8ms.

Tabla 115.PR-06 Reiniciar Nivel modo Desafío

Capítulo 9

Manual de Instalación

En este capítulo se explicarán los pasos necesarios a realizar para lograr la correcta instalación de la aplicación para su posterior uso.

A la hora en la que procedamos a su instalación, nos podemos encontrar tres posibles situaciones, las cuales se explicarán a continuación:

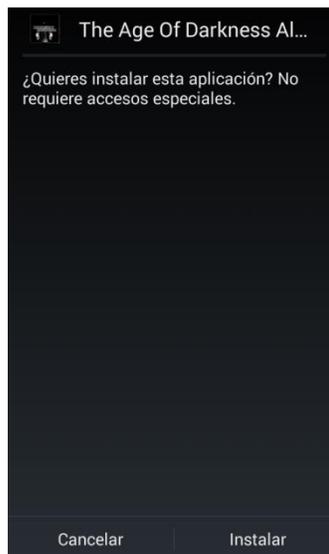
1. Obtención .APK en dispositivo móvil vía externa a “GooglePlay”

Actualmente, debido a que el juego está incompleto, no se encontrará disponible en el “googlePlay” y por ello, la única forma de llevar a cabo su instalación es mediante el uso directo de su ejecutable(.APK). Esta, podrá ser obtenida mediante su envío a través de bluetooth, e-mail, dispositivo físico pendrive...o cualquier otra forma ya sea física o digital de transferencia de archivos.

1.1 Obtener ejecutable .apk a través de algún medio anteriormente descrito.

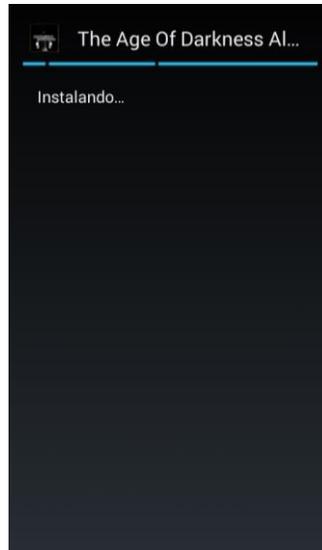
1.2 Ejecutar el archivo **TheAgeOfDarkness.apk**

Tras ejecutarlo, si disponemos de espacio de almacenamiento en el dispositivo (la aplicación no necesita de ningún otro permiso especial), se abrirá una pantalla en la que nos dará las opciones de aceptar la instalación, o por el contrario, la de cancelarla.



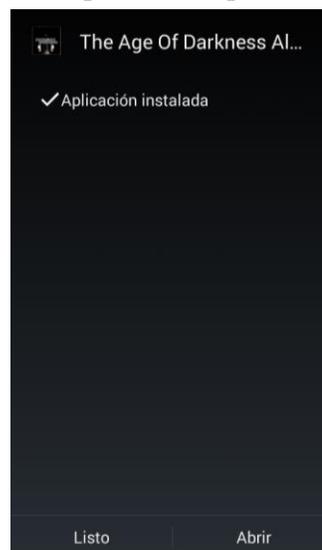
1.3 Instalación del archivo en el dispositivo

Se llevara a cabo el proceso de instalación de la aplicación en el dispositivo, este proceso tardará unos segundos.



1.4 Notificación instalación

Al finalizar el paso anterior, en la misma pantalla aparecerá un mensaje notificando al usuario que la instalación se ha llevado a cabo de manera correcta. En la parte inferior de dicha pantalla se podrán visualizar los botones: Listo o abrir.



- **Listo:** Se cerrará la ventana actual terminando el proceso de instalación.
- **Abrir:** Se cerrará la ventana actual, se terminará el proceso de instalación y se ejecutará la aplicación.

2. Obtención .APK en dispositivo móvil via “GooglePlay”

Tras estar subida la aplicación al “google play”, lo primero será acceder a ella y, tras esto, para facilitar la búsqueda del archivo iremos al buscador, esta búsqueda será realizada mediante una búsqueda por nombre: <<The Dark World - The Age Of Darkness>>.

Si nuestro dispositivo móvil tiene capacidad de almacenamiento suficiente, se llevará a cabo el proceso de instalación de manera automática.

Cuando esta termine se nos notificará y podrá llevarse a cabo la ejecución del videojuego.

3. Obtención .APK en ordenador

En caso de no disponer de dispositivo móvil o que este no disponga de sistema operativo Android, la aplicación podrá ejecutarse mediante la utilización de un emulador Android o desde el mismo Unity3D.

3.1 Emulador

Un ejemplo de emulador podría ser BlueStack, considerado uno de los mejores emuladores de juegos para android en PC.

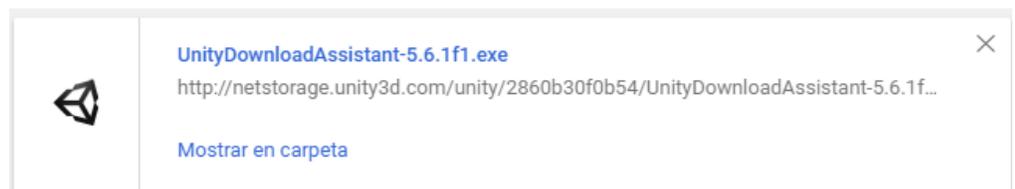
3.2 Unity3D

La versión de Unity3D con la que se ha trabajado ha sido la 5.6.1f, última versión del programa hasta el momento, y dicha versión la podemos encontrar en la página oficial de Unity3D⁷.

Tras descargar el ejecutable, procedemos a ejecutarlo para su instalación.

Se mostrarán a continuación los pasos a realizar en caso de realizarlo mediante un ordenador con Windows.

3.2...1. Ejecutar el archivo descargado



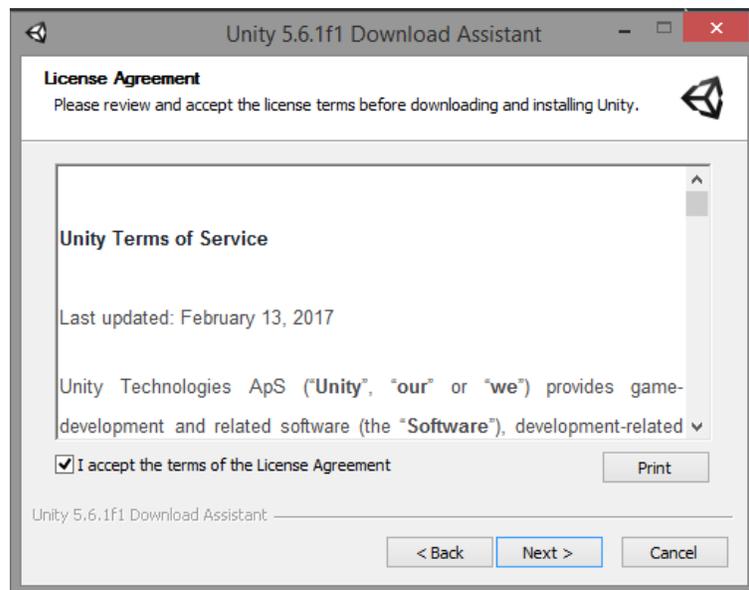
⁷ <https://unity3d.com/es/get-unity/download/archive>

3.2...2. Instalación



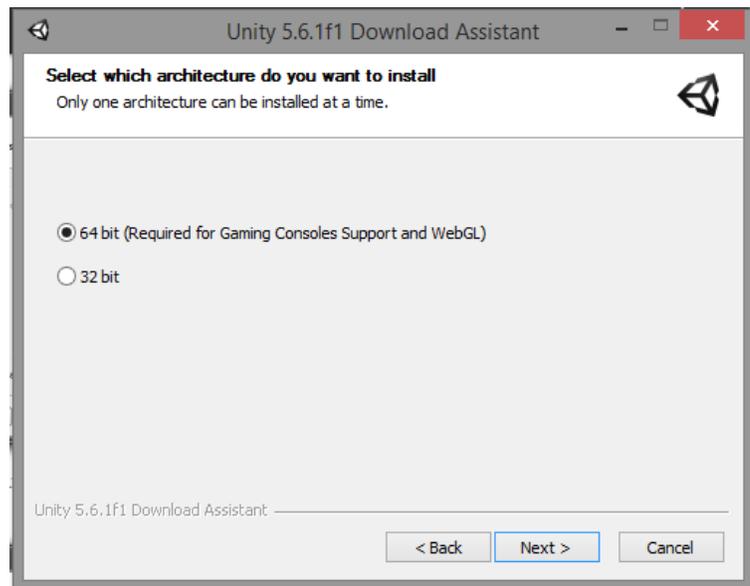
Pulsar Siguiente/Next.

3.2...3. Instalación II



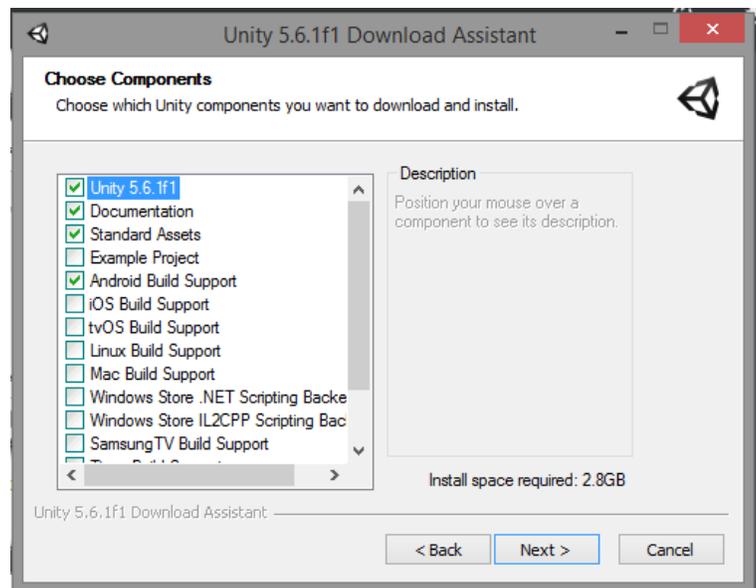
Aceptar la casilla <<Aceptar Términos y Condiciones>>
Pulsar Siguiente/Next.

3.2...4. Instalación III



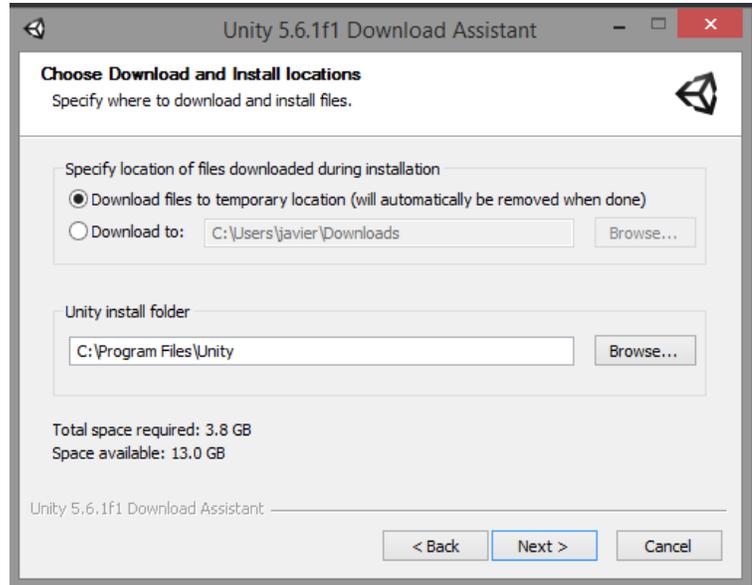
Seleccionar el tipo de procesador que tenga nuestro ordenador, ya sea portátil o de sobremesa (x64/x32)
Pulsar Siguiente/Next.

3.2...5. Instalación IV



Dejar seleccionados por defecto los componentes a instalar que nos marca Unity.
Pulsar Siguiente/Next.

3.2...6. Instalación V

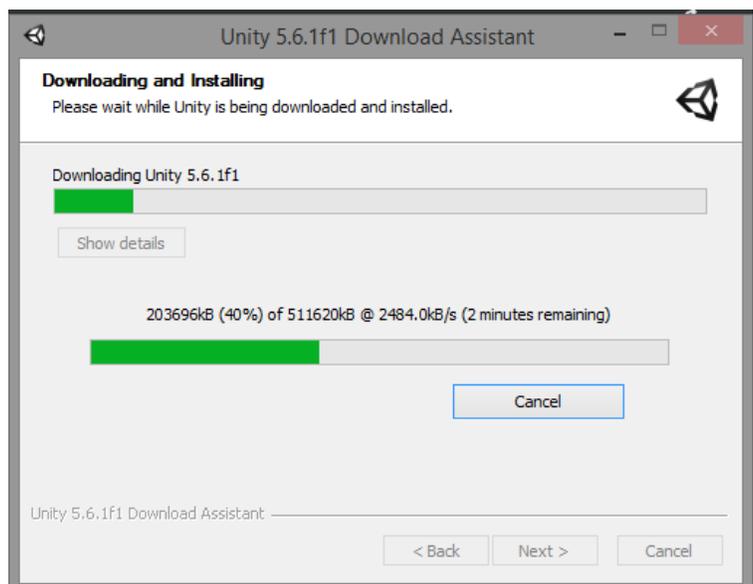


Dejar todo por defecto.

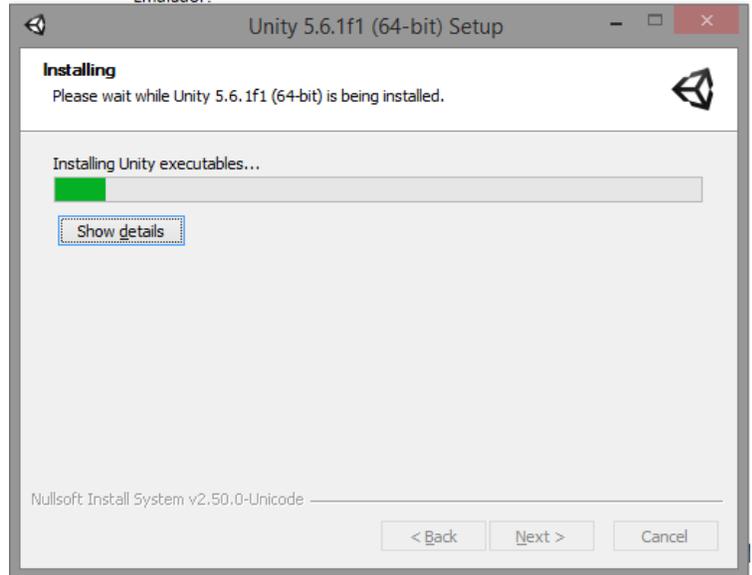
Si se desea instalar Unity en otra ruta distinta a la seleccionada, esto se deberá cambiar en el cuadro de texto inferior de la ventana (-Unity Install Folder-)

Pulsar Siguiente/Next.

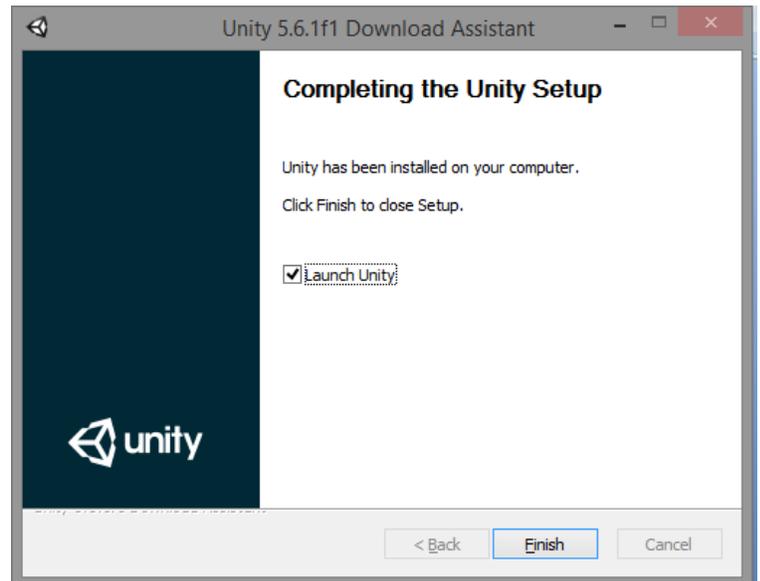
3.2...7. Instalación VI



Proceso de descarga de la instalación de Unity3D en su ordenador, este proceso llevara unos minutos.



3.2...8. Instalacion VII



Tras la finalización de la instalación se podrá lazar directamente Unity3D (dejando marcado el checkBox—Launch Unity) o, si se quiere ejecutar más tarde, desactivar el checkbox y finalizar la instalación.

3.2...9. Tras ejecutar el programa, se podrá acceder al proyecto pulsando el botón Abrir/Open.



3.2...10. Accedemos al CDRom/DVD y seleccionamos la carpeta <<**The Dark World -The Age Of Darkness**>>

3.2...11.Tras abrir el proyecto, se abrirá la interfaz de Unity3D con el juego cargado

3.2...12.A continuación se nos abrirá el proyecto en Unity3D

3.2...13.Comprobar que se ha cargado la escena principal del juego “HomeScene” y ejecutar el botón Play para comenzar el juego.

Unity 5.6.1f1 Personal (64bit) - HomeScene.unity - New Unity Project - Android <DX11 on DX9 GPU>

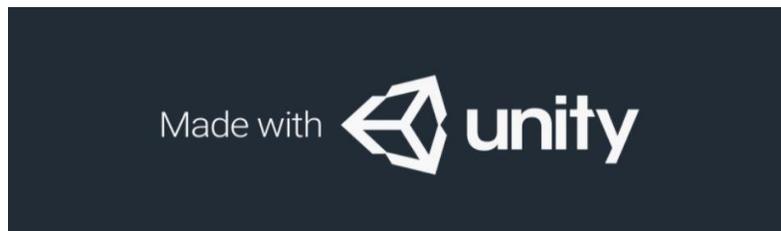


Capítulo 10

Manual de Usuario

En este capítulo se va a proceder a mostrar el funcionamiento de la aplicación.

Lo primero que cualquier usuario se encontrará al ejecutar la aplicación será una pantalla de inicio en la que saldrá el logo de Unity (“SplashScreen”), esta se muestra por defecto debido a haberse desarrollado con el plan gratuito de dicho motor.



1.- HomeScene

Tras esperar unos segundos, se nos mostrará la pantalla inicial del juego, en la que se permitirá al usuario:

- Acceder al menú configuración/ajustes: Permitirá la gestión de la música de fondo del videojuego,
- Acceder al modo de juego Plataforma: Permitirá acceder a dicho modo de juego.
- Acceder al modo de juego Desafío: Permitirá acceder a través de un panel, a la pantalla de selección de nivel.
- Acceder al modo de juego Historia/Campaña: No implementado
- Salir de la aplicación: Permitirá salir de la aplicación.

Resultado:



2.- Menú configuración/ajustes

Se mostrará dicho menú, encargado de “apagar/encender” la música de fondo del videojuego.

Resultado:



Para salir de dicho menú simplemente hay que pulsar el botón táctil: “VOLVER”

3.- MainScene Modo Plataforma

Se mostrará una pantalla, la cual será la encargada de iniciar el videojuego pulsando la tecla central “animada” “PLAY”.

Resultado:



4.- GameScene Modo Plataforma

Se mostrará una pantalla en donde encontraremos a nuestro personaje principal en un estado “parado” esperando a que el usuario pulse la pantalla para comenzar su movimiento, movimiento que ya no parará hasta que el personaje caiga al vacío.

Para saltar entre plataformas solamente hay que volver a tocar la pantalla cuando se quiera realizar dicho salto.

A través de su viaje, irá consiguiendo puntos, estos se podrán obtener mediante la obtención de cristales (3puntos) a lo largo del mapa y, las plataformas heladas de tamaño pequeño que ya que añaden dificultad al jugador darán una pequeña bonificación en la puntuación (1punto).

Si el personaje se precipitará al vacío, aparecería un panel indicándonos que el personaje a perdido y nos encontramos en GameOver, dando las opciones de reinicio o salida.

Resultado Juego:



Resultado GamOver:



3.- Selección Nivel Modo Desafío

Se mostrará una pantalla, con los niveles disponibles a los que el usuario/jugador podrá acceder.

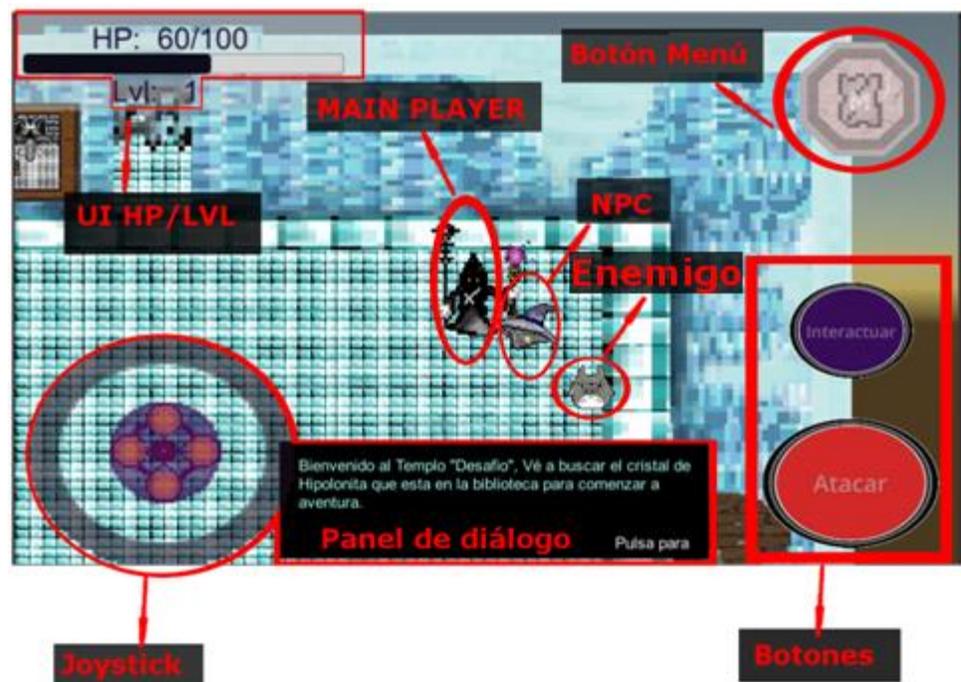
Resultado:



3.- Zone1MainRoom Desafío

Se trató del nivel uno del juego, y es el nivel el cual va a ser mostrado para esta versión del proyecto.

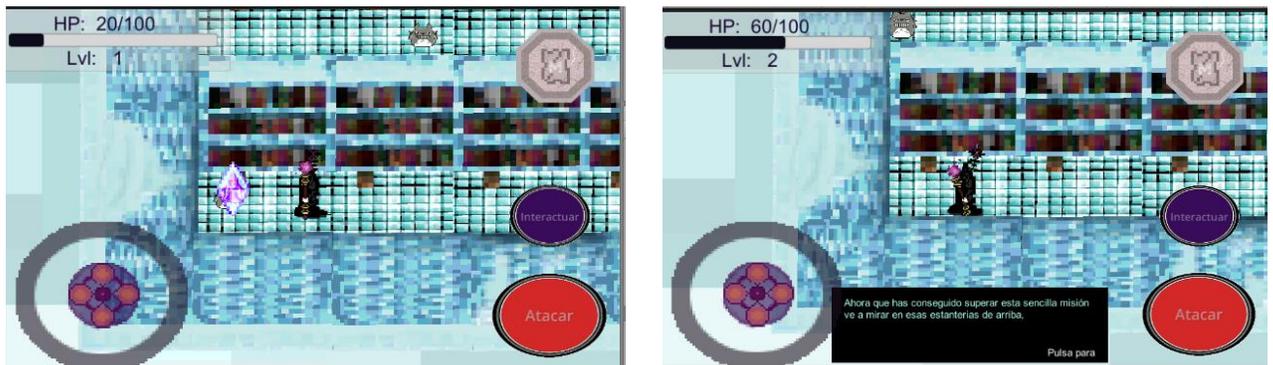
Resultado:



Como podemos ver en la imagen anterior, se pueden distinguir distintos elementos.

- **Interfaz Botones+Joystick:** La interfaz básica de este modo de juego consta de un joystick con el que el usuario podrá moverse por las salas del mapa, dos botones principales (Interactuar/Atacar) los que nos permitirán atacar a nuestros enemigos e interactuar con los elementos de la sala, o al menos, con los que se pueda. Además, a añadir, el botón de la esquina superior derecha nos permitirá desplegar el menú de pausa, aunque el tiempo mientras tanto NO será detenido...o si que cuidado!
- **Interfaz HP/LVL:** Será la interfaz gráfica que nos mostrará tanto la vida total, como la actual de nuestro personaje, esta lo hará mediante una indicación numérica y mediante una forma visual (“Slider”/Barra de vida) y, además, el nivel de esté..
- **Panel de diálogo:** Indistintamente se trate en una misión con activación automática o manual, el panel de diálogo aparecerá en la zona inferior de la pantalla y se pasarán mediante el uso del botón “Interactuar”.

Activación Automática de zona-Misión 1- ItemQuest



Activación Manual:



- **NPC:** Actualmente sólo se encuentra disponible este NPC y será el encargado de proporcionar la primera misión del nivel al jugador.
- **Enemigo:** Actualmente sólo se encontrarán enemigos de este tipo (“Tompli/Totoro”), éstos con un movimiento random.
- **MainPlayer:** Personaje controlado por el usuario, su objetivo será el de completar todos los niveles de desafío, superando sus misiones, tanto lógicas como de acción para ir desvelando los secretos más ocultos de este mundo...

Si nuestro personaje se queda sin puntos de vida, entonces aparecerá una animación de muerte del personaje y aparecerá una pantalla de GameOver.



Capítulo 11

Conclusiones

La realización de este proyecto se ha llevado a cabo tras optar un año por el tfg, pero debido a las circunstancias y, que el tema del TFG, aún siendo elegido por íi, no me motivaba, estaba más orientado a empresas, más concretamente a un taller.

Tras cierto tiempo de deliberación con mis amigos pensando ideas, no se nos ocurría ninguna que me llegara a motivar hasta que pensé que lo que en realidad me motiva y por lo que entré a la carrera fue por los videojuegos, siempre he estado en la posición del jugador por eso pensé que era una buena idea para introducirme en este mundo y, meterme en la piel de aquellos que desarrollaron ó desarrollan juegos, pudiendo conocer ambos lados de mi “pasión” los videojuegos.

Tras ver como se ha ido desarrollando el juego y el resultado final, pienso que ha sido una buena decisión la elección, tanto por la motivación que a ejercido en mí para seguir desarrollandole, como por los conocimientos aprendidos sobre esta área en lo relativo al motor Unity3D y el lenguaje de programación en C#.

Durante el desarrollo del videojuego, he encontrado distintas dificultades, la principal fue el diseño de un videojuego desde cero, ya que, como carecía de cualquier conocimiento más haya del ámbito “jugador” sobre esta área, era un mundo nuevo para mí,

Ese desconocimiento, cabe destacar, que se ha apoyado gran parte en la documentación ofrecida por UNITY3D ayudando en gran parte a implementar el videojuego.

Respecto a la parte más “formal” del proyecto, la documentación, y sus fases de análisis y planificación, se ha apoyado en gran parte en años pasados donde se cursaron las asignaturas correspondientes, como por ejemplo: Análisis de Requisitos.

El resultado final del proyecto no supera las expectativas que me había formado, esto puede ser debido a qué como jugador que soy, comencé el videojuego con una idea (En un principio sólo se pensó realizar un único modo de juego) pero está, según avanzaba el desarrollo fue ampliándose hasta el punto de evolucionar a un multi-juego. Esto me pasó con todo lo relacionado al juego, y al final muchas de las cosas que he implementado o que he intentado implementar no han llegado a esta primera versión “final” del juego ya que, como eran tantas cosas no se ha podido llegar a la realización o puesta en marcha de todas.

Sin embargo, aunque no supere las expectativas que tenía puestas, su realización ha sido muy positiva para mí , no sólo por haberme puesto “en el otro lado” de los videojuegos, sino porque he adquirido conocimientos sobre el área y, porque, aunque esta primera ver-

sión no haya superado mis expectativas, tengo esperanzas de que en un futuro con dedicación llegaré a ese ideal que tengo como juego para así poder compartirlo con mucha gente la cuál opina como yo y puedan disfrutar como yo en su día disfrute con aquellos juegos que tanto me gustaban, y me gustan.

11.1 Futuras mejoras

En esta sección se muestran algunas de las futuras mejoras que se pretenden hacer en el proyecto.

Voy a comentar alguna de las mejoras más importantes que veo para implementar en el futuro, porque

- **Modo Historia:** Implementación de un modo historia en el que la jugabilidad y la propia historia estén a la par, es decir, que se caracterice por esas dos características. Esto será todo un reto porque el ideal que tengo es el de un mundo de grandes dimensiones el cuál pueda recorrer el usuario de forma libre, pero con el objetivo de seguir la misión principal del juego o submisiones que nos vayamos encontrando.
- **Niveles desbloqueables:** En modo desafío, se implementará en un futuro el uso de niveles desbloqueables.
- **Enemigos:** Se han diseñado gran variedad de enemigos, así como Bosses (Jefes de zona) pero, al final, para esta versión solamente se ha llegado a implementar un enemigo con movimiento Random, se espera implementar todos para un futuro y añadirles distintas funcionalidades dependiendo del enemigo.

Cabe destacar que algunos de los enemigos pretenden ser un “guiño” a otros juegos.

- **Clasificaciones:** En el panel de selección Modo desafío vemos como hay una opción llamada “Clasificación” está mostrará en un futuro los tiempos en los que se han realizado los distintos niveles del modo desafío, otorgándole un carácter aun poco competitivo.
- **Monturas:** Se espera que en el modo plataformas, al llegar a cierta puntuación se vayan desbloqueando monturas nuevas con las que nuestro personaje intentará llegar más lejos, estas tendrán características distintas. (Ejemplo más fuerza de salto, más velocidad de movimiento...)

- **Sistemas de logros GooglePlay:** Para ello falta mucho, pero en un futuro, cuando se tenga subida la aplicación al GooglePlay se esperará añadir logros para anexionar más el juego con dicha plataforma. Además, porque me parece algo muy llamativo en un juego y algo que hace que alguien vaya más allá intentando conseguir dicho logro o logros, aumentándole las ganas de jugar.
- **Sonido:** Se pretende añadir más efectos de sonido y distintas músicas de fondo para los distintos modos de juego, además de controlar todo con el menú configuración/ajustes.
- **Unión:** Con la creación de un tercer juego se espera unificar en mayor medida los controladores del juego.

Capítulo 12

Bibliografía & Webgrafía

Este capítulo contiene tanto la bibliografía como la webgrafía utilizada a lo largo del desarrollo de este proyecto.

Cabe destacar que la gran mayoría de información, ya sea a través de manuales o tutoriales se ha obtenido a través de la página web de Unity3D.

Bibliografía:

- Nicolás Arroja Landa. Unity. Diseño y programación de videojuegos. ISBN 978-987-1857-81-4
- Apuntes Miguel A.Martínez. Gestión, elicitación, documentación de Requisitos. 2014
- Apuntes Francisco J.González Cabrera. Planificación de proyectos, estimación de costes, esfuerzo y tiempos de un nuevo proyecto software 2012.
- Apuntes Francisco J.González Cabrera. Proceso de desarrollo del software. 2013

Webgrafía:

- Joaquín Casas Albertos [Estudiante UVA] - Organización Documentación.
- Joaquín Casas Albertos [Estudiante UVA]- Arquitectura de Unity3D.
- Unity Technologies. Unity Manual [en línea]. [Última consulta 13/07/2017]
- Nodician. Como hacer música para un juego rpg con fl studio 12. [En línea]. [Última consulta 20/06/2017]. Disponible en: <https://www.youtube.com/watch?v=aDt2OIE5UTU>
- Mundoalexo. CREA MÚSICA PARA TUS VIDEOJUEGOS DE MANERA FÁCIL CON FL STUDIO (CUALQUIER VERSIÓN) [En línea].m [Última consulta 20/06/2017] Disponible en: <https://www.youtube.com/watch?v=sS05hPiBqbo>
- Blzgz. Crear RPG en Unity3D. [En línea]. [última consulta 24/06/2017] Disponible en: <https://www.youtube.com/playlist?list=PLNEYXN2JPIqxIHPpoCW7qIw257DEjpU0W> [Lista de reproducción: 15 vídeos]
- Blzgz. Cómo hacer un inventario en Unity3D. [En línea]. [Última consulta 24/06/2017] Disponible en: <https://www.youtube.com/playlist?list=PLNEYXN2JPIqyTPmAakvelc-JBs8eLstoZ> [Lista de reproducción: 12 vídeos]

- Blzksz. Unity: Introducción y herramientas. [En línea]. Última consulta 24/06/17] Disponible en: <https://www.youtube.com/playlist?list=PLNEYXN2JPIqwHEHyjR5HXZvraYcqeq6II>
[Lista de reproducción: 7 vídeos]
- Foro para iniciarse en el mundo de los sprites, 2D. [En línea]. [Última consulta 14/06/17] Disponible en: <https://wahackforo.com/>
- PixelArtTutorial [En línea]. [Última consulta 14/06/17] Disponible en: <http://makegames.tumblr.com/post/42648699708/pixel-art-tutorial>
- MikeGeigTV. Revisiting 2D Parallax Scrolling Backgrounds in Unity [En línea] [Última consulta: 20/06/2017] Disponible en: <https://www.youtube.com/watch?v=DIQFhEo1C8c>
- Hector Pulido. Pokemon en Unity3D- 8.6 Sistema de Combate-Por fin animaciones!. [En línea] [Última consulta 20/06/2017] Disponible en: <https://www.youtube.com/watch?v=vdvOPMVq9Ow>
- Hector Pulido. Crear un juego como pokemon. [En línea][Última consulta 20/06/2017]. Disponible en: <https://www.youtube.com/playlist?list=PLHdk97dabfvwnhZI64A4A952NixnvDZ61>
[Lista de reproducción: 17 vídeos]
- Hector Pulido. Sistema de inventario Unity3d [En línea][Última consulta 20/06/2017]. Disponible en: <https://www.youtube.com/playlist?list=PLHdk97dabfvznNPk1N6Hxps925X0ypFRR>
[Lista de reproducción: 7 vídeos]
- BrainStrom Games. Lets Make... Pokemon in Unity! - Episode 1 - Basic Setup World/Character. [En línea][Última consulta 25/07/2017] Disponible en: https://www.youtube.com/watch?v=_PagngtnZ3U -[Consta de 7 videos adicionales]
- Kaisirak. Unity 2D Tilemapper. [En línea] [Última consulta 25/07/2017] Disponible en: https://www.youtube.com/watch?v=_x0bMTxP7Yw
- Hagamos Videojuegos. Creando un juego de tipo “rogue” [En línea] [Última consulta 25/07/2017] Disponible en: <https://www.youtube.com/playlist?list=PLREdURb87ks2qkD9svvlIwYwN35FZ3Afv>
[Lista de reproducción: 15 vídeos]
- Hagamos Videojuegs. Creando un juego “infinite Runner” en 2D con unity 4.3 (Tutorial) [En línea] [Última consulta 25/07/2017] Disponible en: https://www.youtube.com/playlist?list=PLI8ExHydRGsXbeob9-NqPE_6NcCIZyZfE
[Lista de reproducción: 25 vídeos]
- Julio Cesar. Tutorial Unity 2D - Clonando a Flappy Bird. [En línea] [Última consulta 25/07/2017] Disponible en:

https://www.youtube.com/playlist?list=PLI8ExHydRGsXbeob9-NqPE_6NcCIZyZfE
[Lista de reproducción: 8 vídeos]

- Julio Cesar. Tutorial Unity 5 - Game Physics. [En línea] [Última consulta 25/07/2017] Disponible en:
<https://www.youtube.com/playlist?list=PLI8ExHydRGsWzh9rmMNhfhScvZjpiy6Y4>
[Lista de reproducción: 9 vídeos]
- Unity. Tutorials - 2D Roguelike Project. [En línea] [Última consulta 25/07/2017] Disponible en: <https://www.youtube.com/playlist?list=PLX2vGYjWbI0SKsNH5Rkpxvxl1dPE0Lw8F>
[Lista de reproducción: 14 vídeos]
- Devin Curry. Unity 5 Mobile Joysticks Tutorial - Touch Input 2D Spaceship Controller [En línea] [Última consulta 25/07/2017] Disponible en:
<https://www.youtube.com/watch?v=nGYObojmkO4>
- Comparativa Unity y Cocos2D. [En línea] [Última consulta 25/07/2017] Disponible en:
<https://www.yeeply.com/blog/comparativa-unity-cocos2d/>
- Comunidad visitada GameDev [En línea] [Última consulta 25/07/2017] Disponible en:
<http://www.gamedev.es/>
- El modelo COCOMO [En línea] [Última consulta 25/07/2017] Disponible en:
<http://www.sc.ehu.es/jiwdocoj/mmis/cocomo.htm>
- Metodología Waterfall process. [En línea] [Última consulta 25/07/2017] Disponible en:
<http://www.umsl.edu/~hugheyd/is6840/waterfall.html>
- Metodologías de desarrollo de videojuegos. [En línea] [Última consulta 25/07/2017] Disponible en:
https://senaintro.blackboard.com/bbcswebdav/institution/524143_tvirtual/contenidos/OA/Fase4/Metodologias%20de%20desarrollo/adjuntos/MetodologiasDeDesarrollo.pdf

Capítulo 13

Glosario

Este capítulo contiene el vocabulario técnico utilizado a lo largo del documento:

Asset: Se tratán de paquetes de recursos, estos pueden tratarse de cualquier contenido que componga un juego realizado en Unity3D.

Boss: Se refiere a los jefes de zona o de nivel de los distintos juegos en los que se use, pero, a grandes rasgos, en general se refiere a un enemigo el cual es superior a todos los de la zona.

Un ejemplo, es, en el apartado 6.6 podemos encontrar 3 tipos diferentes de jefes, uno de los cuales esta caracterizado por disponer de 2 espadas y 3 escudos, este, en un futuro, se caracterizará por disponer de distintas fases, dificultando el juego.

Demo: Se trata de una versión-demostración de un juego distribuido con fines promocionales, es decir, que no tiene los funcionalidades completas y que, se suelen lanzar para ayudar al consumidor a decidirse por la compra final del juego.

Frame-by-frame: Literalmente significa Imagen-por-imagen. Y consiste en la superposición de imágenes de forma lo suficientemente rápida como para que está, de sensación de movimiento.

GUI/UI: Interfaz de Usuario Gráfica/ Interfaz de Usuario.

HUD: Significa: Heads Up Display y su función ee la de proporcionar información relativa al jugador.

HP: Acrónimo de Health Points (Puntos de salud)

Level/lvl: Se refiere al nivel, mas concretamente en este caso, se utilizará para referirnos al nivel del personaje.

NPC'S: Sus siglas significan Non-Player Character y se refiere a aquellos personajes que no son jugables, que son controlados por el juego.

Prefab: Se refiere a las “plantillas” de los objetos, es decir, objetos los cuales se quiere que

mantengan propiedades específicas para su posterior utilización.

Smartphone: Dispositivo inteligente con gran capacidad de almacenamiento de datos y realizar actividades como si se tratase en un mini-ordenador, además, actualmente, con una conexión casi permanente a la red.

SplashScreen: Pantalla previa al inicio del juego la cuál dura unos segundos, está, por defecto, en este caso al tratarse de la versión gratuita de Unity3D mostrará un logo de los mismos.

Sprite: Se tratan de mapas de bits en 2D, estos, solo poseen anchura y altura y están compuestos por pixeles.

Spritesheet: Agrupación de distintos sprites en un sprite llamado spritesheet, con esto se consigue ahorrar memoria y nos facilitará la búsqueda de sprites.

Videojuego AAA: Los videojuegos triple A son aquellos desarrollados por grandes compañías de videojuegos y que cuentan con grandes presupuestos para su producción y promoción, nivel muy alto de gráficos... generando gran demanda entre los usuarios. (AAA) Hace referencia a un término inglés, el cuál es muy común usado para calificar exámenes siendo AAA matricula de honor.

Videojuego género Plataformas: Videojuego caracterizado por tener que caminar/saltar/correr...sobre una serie de plataformas las cuales se encuentran suspendidas del aire con el objetivo de recoger distintos objetos para completar el juego y cuya caída solo depara la muerte. Este tipo de videojuegos suele usar vistas caracterizadas por un movimiento horizontal.

Videojuego género Action-RPG: Videojuego caracterizado por la posibilidad de controlar a un único personaje en tiempo real y suelen estar centrados en el combate y la acción. Puede tratarse de mundos abiertos, cuya jugabilidad no es lineal y donde mediante un botón de ataque nuestro personaje realizara la acción de atacar con espada, arco...

Videojuego género RPG: Videojuego caracterizado por ser parecido al anterior pero con la peculiaridad que los combates serán por turnos, esto es, porque originalmente fueron basados en los juegos de rol de tableros, como por ejemplo dragones y mazmorras,

La gran mayoría de la gente que juega este tipo de juego lo hace por su historia, después la exploración (descubrir los entresijos de la historia del universo de dicho juego), esta exploración suele llevarse a cabo mediante las misiones secundarias repartidas a lo largo del mundo.

Videojuego Indie/independiente: Son juegos creados por individuos o grupos pequeños que no cuentan con gran capital de inversión ni apoyo financiero externo.

A menudo se basan en la innovación y su distribución por medios no físicos, como por ejemplo: GooglePlay. Estos, en los últimos han incrementado debido principalmente a los nuevos métodos de distribución en línea y herramientas de desarrollo.

Capítulo 14

Contenido del CD-ROM

En este capítulo se especificará el contenido del CD-ROM/DVD:

- Titulo_TFG.pdf: Portada del proyecto.

- Memoria_TFG.pdf: Memoria del proyecto

- Diagramas: Contendrá los diagramas del documento en mayor tamaño para mejorar su visualización en el caso de que hiciera falta.

- Diseño: Contendrá todos los elementos relativos al proceso de Diseño del proyecto
 - Animations: Animaciones del juego

 - Logo: Logo del juego

 - Sprites: Todos los sprites creados

 - Bocetos: Bocetos sobre las interfaces de usuario del juego

- Software: Software del proyecto.