



**Universitat Oberta
de Catalunya**

Diseño e implementación de un juego para plataforma Android: Jim Adventure

Antonio Salvador Ramirez Torreno

Ingeniería Técnica de Informática de Gestión

Consultor: Helena Boltá Torrell, Jordi Almiral López

8 de Enero de 2015



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

FICHA DEL TRABAJO FINAL

Título del trabajo:	Jim Adventure
Nombre del autor:	Antonio Salvador Ramírez Torreno
Nombre del consultor:	Helena Boltá Torrell
Fecha de entrega (mm/aaaa):	01/2015
Área del Trabajo Final:	Desarrollo Aplicaciones Dispositivos Móviles Android
Titulación:	<i>Ingeniería Técnica de Informática de Gestión</i>
Resumen del Trabajo (máximo 250 palabras):	
<p>En este proyecto Fin de Carrera se ha desarrollado un juego de los llamados runners y clasificado como Arcade para plataforma Android®. La peculiaridad de estos juegos es que el personaje está en continuo movimiento. De manera que el usuario, para completar las misiones, tiene que controlar dos posibles movimientos: que el personaje salte, o que el personaje ruede agachándose. Haciendo uso de estos dos movimientos, el usuario tiene que completar las distintas misiones recogiendo los objetos indicados en cada nivel.</p>	

Abstract (in English, 250 words or less):

It's a game for Android© platform. This kind of game is called Runners and it's classified as Arcade Game. The particularity of this kind of game is the main character is continually running and the user has to use main character's abilities as jump or roll in order to catch all the objects requested to pass each level in a specific time.

Palabras clave (entre 4 y 8):

Game, Android, Platform, Runner

Índice

1. Introducción	1
1.1 Contexto y justificación del Trabajo	1
1.2 Objetivos del Trabajo	3
1.3 Videojuego propuesto.....	4
2. Análisis, Diseño e Implementación.	5
2.1 Análisis.	5
2.1.1 Enfoque y método seguido.	5
2.1.2 Planificación del Trabajo.....	6
2.1.4 Diagrama de actividad del sistema.....	9
2.2 Diseño.....	10
2.2.1 Prototipo.	10
2.2.1.1 Pantalla menú principal.	10
2.2.1.2 Menú New Game o Continue (continuar una partida).....	10
2.2.1.3 Escenario 1.	11
2.2.1.4 Escenario 2.	11
2.2.1.5 Opciones.	12
2.2.1.6 New Record.	12
2.2.1.7 Ranking.	13
2.2.2 Prototipo alta fidelidad.....	14
2.2.2.1 Menú principal.....	14
2.2.2.2 Menú New Game o Continue (continuar una partida).....	15
2.2.2.3 Escenario 1.	15
2.2.2.4 Escenario 2.	16
2.2.2.5 Opciones.	16
2.2.2.6 New Record.	17
2.2.2.7 Ranking.	17

2.3 Implementación.	18
2.3.1 Objetivo.	18
2.3.2 Escenas.	18
2.3.3 Scripts.	21
2.3.4 Monobehaviour.	25
2.3.5 Ficheros.	28
2.3.6 Requisitos del Sistema.	28
2.3.7 Pruebas.	29
3. Introducción a los programas utilizados	33
3.1 InkScape.	33
3.2 Unity3D.	37
3.2.1 Características.	37
3.2.2 Interfaz de usuario.	38
4. Conclusiones.	44
5. Cambios con respecto al proyecto inicial.	45
6. Líneas Futuras.	46
5. Glosario	47
6. Bibliografía	49
7. Anexos	51

Lista de figuras

Figura 1 - Grafica evolución ventas smartphones.....	1
Figura 2 - Grafica evolución mercado OS smartphones.....	2
Figura 3 – Comparación 2º trimestre S.O. instalados en smartphones.....	2
Figura 4 - Fases que componen la metodología sum	5
Figura 5 - Diagrama de gantt.....	6
Figura 6 - Diagrama actividad del sistema	9
Figura 7 - Sketch - portada	10
Figura 8 - Sketch - new game	10
Figura 9 - Sketch - escenario 1	11
Figura 10 - Sketch - pergamino.....	11
Figura 11 - Skecth – Escenario 2	11
Figura 12 - Sketch - Options.....	12
Figura 13 – Sketch – new personal record	12
Figura 14 - Sketch - ranking	13
Figura 15 - Prototipo - menú principal	14
Figura 16 - Prototipo - menú new game o continue.....	15
Figura 17 - Prototipo -escenario 1	15
Figura 18 - Prototipo - escenario 2	16
Figura 19 - Prototipo - options	16
Figura 20 - Prototipo - personal records	17
Figura 21 - Prototipo - ranking.....	17
Figura 22 - Prototipo - pergamino	18
Figura 23 - Producto final - portada.....	19
Figura 24 – Producto final - elección	19
Figura 25 - Producto final - escena 1	20
Figura 26 - Producto final - new record.....	20
Figura 27 - Producto final - level up	20

Figura 28 - Producto final - opciones.....	21
Figura 29 – Producto final - credits.....	21
Figura 30 - MonoBehaviour - orden ejecución de funciones.....	27
Figura 31 - Inkscape - interfaz usuario	34
Figura 32 - Inkscape - bloques.....	34
Figura 33 - Inkscape - lienzo con personaje	35
Figura 34 - Inkscape - personaje principal.....	36
Figura 35 - Inkscape - plataforma	36
Figura 36 - Unity 3d - project wizard.....	38
Figura 37 - Unity3d - distribución ventanas	39
Figura 38 - Unity 3d - ventana inspector	40
Figura 39 - Unity3d - ventana project (proyecto)	41
Figura 40 - Unity3d - ventana game	42
Figura 41 - Unity3d - build project.....	43

1. Introducción

1.1 Contexto y justificación del Trabajo

La venta de Smartphones, o terminales inteligentes, se incrementa cada año. Según un estudio de la consultora IDC [1] el crecimiento, en el segundo trimestre del 2014, ha sido de un 25.3% con respecto al mismo periodo del año anterior, sobrepasando las 300 millones de unidades vendidas.

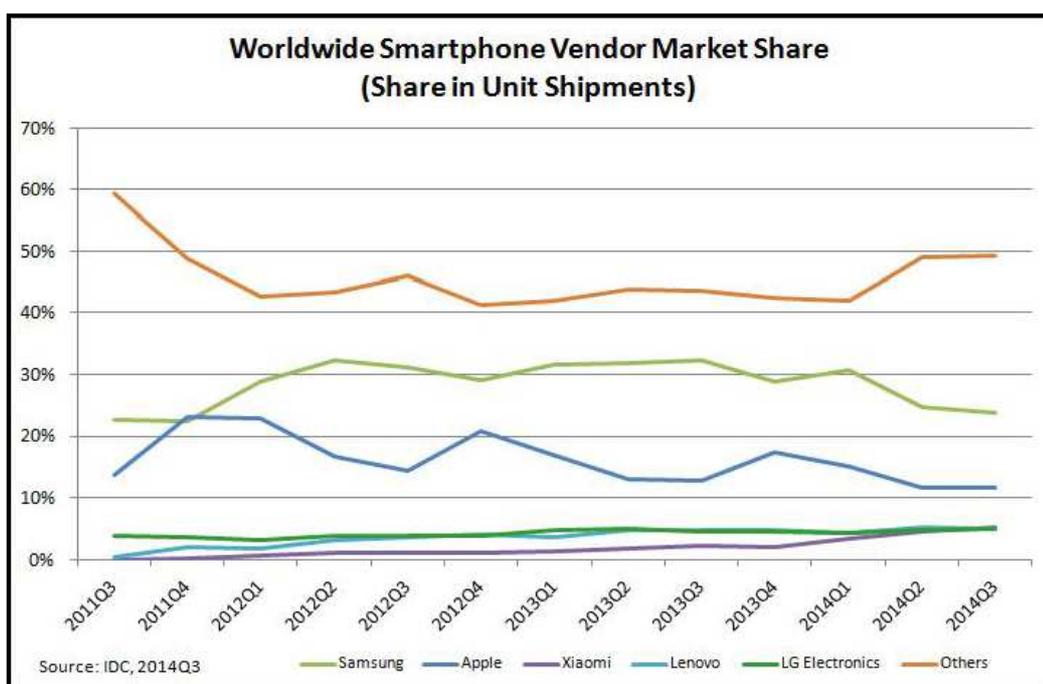


FIGURA 1 - GRAFICA EVOLUCIÓN VENTAS SMARTPHONES

Un 99% de estos terminales están controlados por los siguientes sistemas operativos:

- Android®, creado por Google
- iOS®, que gobierna los dispositivos de Apple
- Windows Phone®, el sistema operativo móvil de Windows
- Black Berry OS®

De entre los sistemas operativos anteriores Android® es el rey indiscutible estando instalado en más del 80% de los Smartphones del mercado, tal como se muestra en el estudio realizado por IDC [2] en el segundo trimestre de 2014.

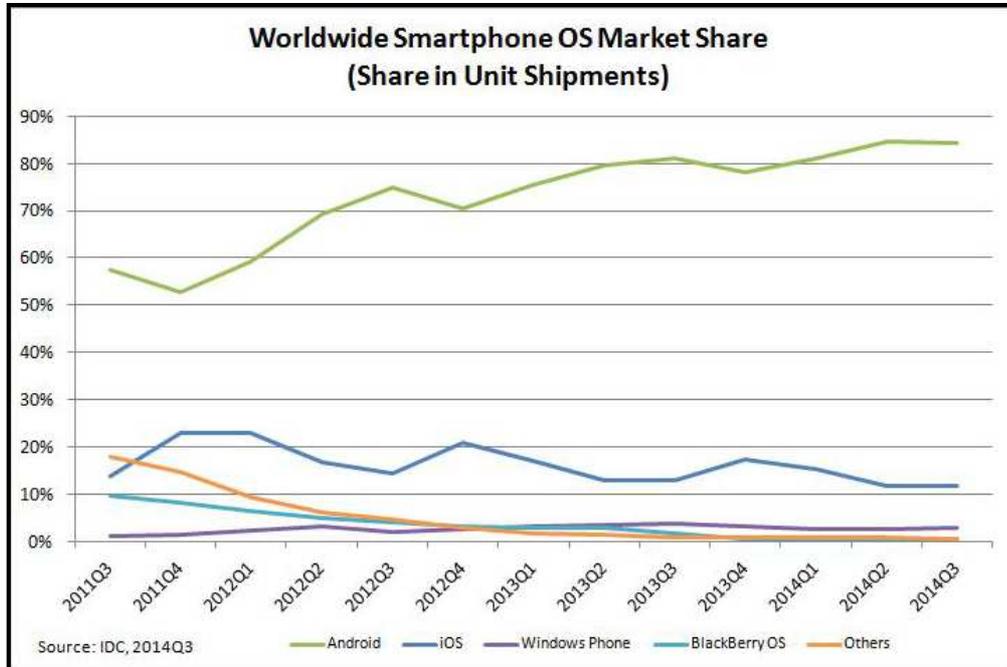


FIGURA 2 - GRAFICA EVOLUCIÓN MERCADO OS SMARTPHONES

Otro sector en alza es la industria del videojuego. Gracias al avance en cuanto a hardware en Pcs, videoconsolas, etc... y al incremento de juegos de una mayor calidad tanto técnica, como visual y mejoras en la interacción con los usuarios, la industria del videojuego ha superado en ingresos a otras disciplinas artísticas como el cine y la música juntas.

Period	Android	iOS	Windows Phone	BlackBerry OS	Others
Q2 2014	84.7%	11.7%	2.5%	0.5%	0.7%
Q2 2013	79.6%	13.0%	3.4%	2.8%	1.2%
Q2 2012	69.3%	16.6%	3.1%	4.9%	6.1%
Q2 2011	36.1%	18.3%	1.2%	13.6%	30.8%

FIGURA 3 – COMPARACIÓN 2º TRIMESTRE S.O. INSTALADOS EN SMARTPHONES

Teniendo en cuenta estos dos factores, mayor incremento de usuarios con terminales Smartphones, y crecimiento de los videojuegos, se puede afirmar que el desarrollo de aplicaciones de ocio para plataformas móviles, y concretando para Android®, tiene una gran salida profesional, por lo que aprender a crear aplicaciones para esta tecnología puede abrir muchas puertas profesionales.

1.2 Objetivos del Trabajo

El objetivo final del trabajo es crear un videojuego para el sistema operativo Android®. El tipo de juego sería de los llamados Runner, donde el jugador controla a un personaje que está continuamente corriendo y evitando obstáculos, recogiendo objetos, saltando de plataforma en plataforma, con el fin de pasar de nivel. Se trataría de un juego 2D.

Para alcanzar este objetivo se necesitará:

- Aprender a utilizar una serie de herramientas para realizar las diferentes tareas:
 - Unity3d® como motor gráfico
 - Gimp 2.0® como editor gráfico
 - InkScape® como editor/generador vectorial.

- Familiarizarse con los términos usados para la creación de un juego como son Sprites, Tiles, animación de Sprites, assets, prefabs, texturas, física de los objetos, inteligencia artificial, gestión de colisiones etc...

- Diseño de escenarios.

- Diseño de personajes y animaciones.

- Creación de scripts para controlar al personaje, flujo de escenarios, IA de enemigos, etc...

- Creación menús y transiciones entre escenarios.

- Añadir funcionalidades sociales: Rankings, Logros.
- Crear la distribución para Android® del producto final.

1.3 Videojuego propuesto.

El juego propuesto es un videojuego de los llamados runners y clasificado como Arcade para plataforma Android®.

El personaje principal, Jim, tiene que avanzar por los diferentes niveles del juego. Para superar cada uno de los niveles hay que recoger una serie de objetos que se indican al comienzo de cada uno de los niveles.

Por tratarse de un juego runner, el personaje está en continuo movimiento a una velocidad que el usuario no puede modificar. Los controles que maneja el jugador son dos:

- Salto para moverse entre las plataformas a distintas alturas y para esquivar los obstáculos.
- Rodar agachándose, para esquivar obstáculos.

Estos controles se realizan a través de la pantalla táctil del dispositivo Android®. Pulsando en la parte derecha de la pantalla se salta, y pulsando en la parte izquierda se rueda.

2. Análisis, Diseño e Implementación.

2.1 Análisis.

Previo al diseño de la aplicación se realizó una encuesta para saber exactamente los usos, gustos, ver ANEXO I : ENCUESTA

2.1.1 Enfoque y método seguido.

Para la creación de este videojuego se va a utilizar la metodología SUM [4], dado que es muy efectiva para proyectos cortos y equipos pequeños.

Esta metodología adapta para videojuegos la estructura y roles de SCRUM.

El ciclo de vida se divide en fases iterativas e incrementales que se ejecutan en forma secuencial con excepción de la fase de gestión de riesgos, que se realiza durante todo el proyecto.

Las fases en las que se divide son:

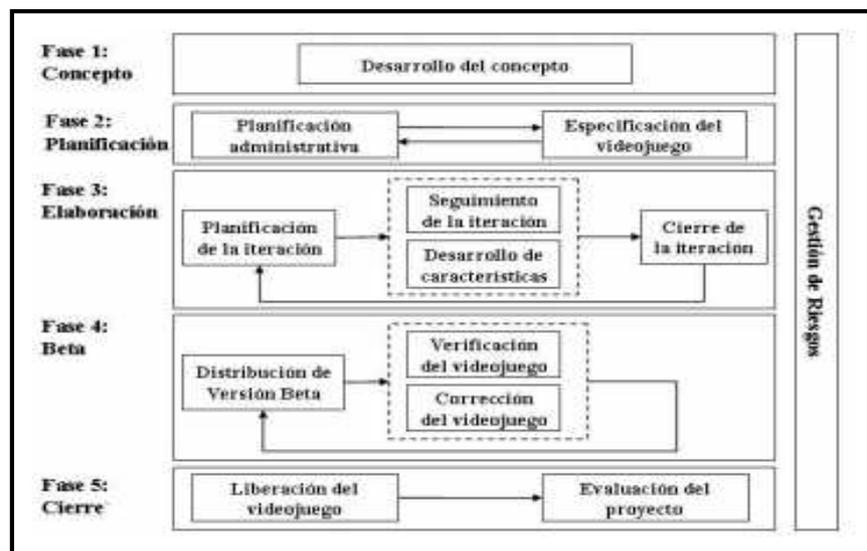


FIGURA 4 - FASES QUE COMPONEN LA METODOLOGÍA SUM

- Concepto: Definición de concepto del videojuego, de los elementos del juego y técnicos. (Esta fase no existe en metodología SCRUM).

- Planificación: Esta fase tiene dos objetivos principales, uno es planificar el resto de las fases y el otro especificar las características a implementar del videojuego. La planificación que se obtiene en esta fase es flexible ya que en

cada iteración se puede modificar para adaptarse. (Se corresponde con la etapa PRE-GAME de Scrum).

- Elaboración: En esta fase se implementa el videojuego. Se trabaja de forma iterativa para obtener una versión ejecutable al final de cada iteración. Esto permite evaluar el avance del proyecto, realizar cambios y tomar decisiones. (Se corresponde con la etapa GAME de Scrum).

- Beta: Se evalúa y ajusta los distintos aspectos del videojuego. (Se corresponde con la etapa POST-GAME de Scrum).

- Cierre: Es la fase en la que se entrega la versión final del juego.

- Gestión de riesgos: Se realiza durante todo el proyecto debido a que los problemas pueden aparecer en cualquier fase.

2.1.2 Planificación del Trabajo.

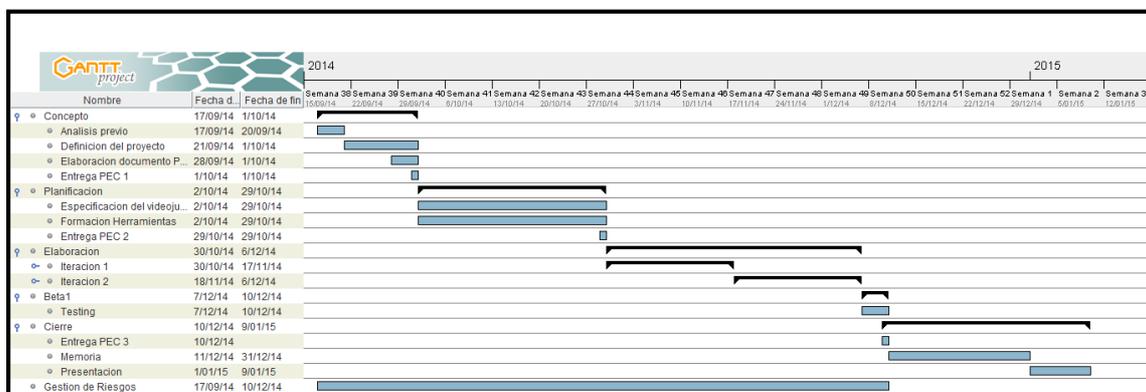


FIGURA 5 - DIAGRAMA DE GANTT

En la planificación están especificadas las distintas fases del proyecto, incluyendo las entregas de las distintas PECs y la generación de documentación. A groso modo se hacía una media de 3 horas, 6 días a la semana.

A continuación se detallan tareas de las distintas fases:

1- Fase Concepto:

1.1 – Análisis previo: Investigación sobre las distintas alternativas para la generación del proyecto, como herramientas, motores gráficos, etc...

1.2 – Definición del proyecto: Se especifica el tipo de aplicación que se quiere construir, el género y el alcance a priori de lo que se quería hacer. Elección de herramientas y motor gráfico a utilizar. Se ha tenido en cuenta casi 60 motores gráficos [3] para este proyecto decantándose por Unity 3d© versión 4.0, ya que ofrece un motor potente para la creación de videojuegos 2D, exportación a multitud de plataformas, una gran documentación disponible tanto en la red como en publicaciones y foros muy activos donde resolver dudas.

Además de esta herramienta se han investigado las distintas utilidades de dibujo gratuitas que pueden ayudar a llevar el proyecto a buen puerto. En este caso, debido a la gran cantidad de tutoriales y manuales se ha optado por Gimp 2.0© y por InkScape©.

2- Fase Planificación:

2.1 – Especificación del videojuego: se recogen las características a implementar, donde se concretan los planes y características definidas.

2.2 – Formación Herramientas: aprendizaje de las distintas herramientas necesarias para la creación del videojuego:

2.3.1 – Unity 3D©: En esta fase aprenderá las funcionalidades, métodos y objetos del motor de Unity 3d©, así como la programación de scripts en C# uno de los tres lenguajes que soporta, junto con Boo y una variante de javascript denominada UnityScript©.

2.3.2 – InkScape©: editor de gráficos vectoriales profesional para Windows, Mac Os X y Linux. Es un software libre y de código abierto.

2.3.3 – Gimp 2.0©: programa gratuito de dibujo, composición de imágenes y retoque fotográfico. Es software libre y multiplataforma (Windows, Mac y Linux).

3- Fase Elaboración: Es la parte de la implementación del juego. Se dividió en dos interacciones.

3.1 – *Iteración 1*: Modelado de personajes, escenarios, enemigos, etc... que ayudan a la creación de una primera versión del juego que se presentara al cliente para la validación, corrección y mejoras que pueda tener.

3.2 – *Iteración 2*: Mejoras de la versión obtenida en la Iteración 1.

4 – Fase Beta: Esta fase se utilizara para últimos ciclos de testeo del software, para su refinamiento, corrección de errores y últimos retoques antes de la entrega como producto final.

5- Cierre: Generación del producto final, con instaladores y documentación (manual del usuario). Comienzo de la memoria y creación de la video-presentación del proyecto.

2.1.4 Diagrama de actividad del sistema.

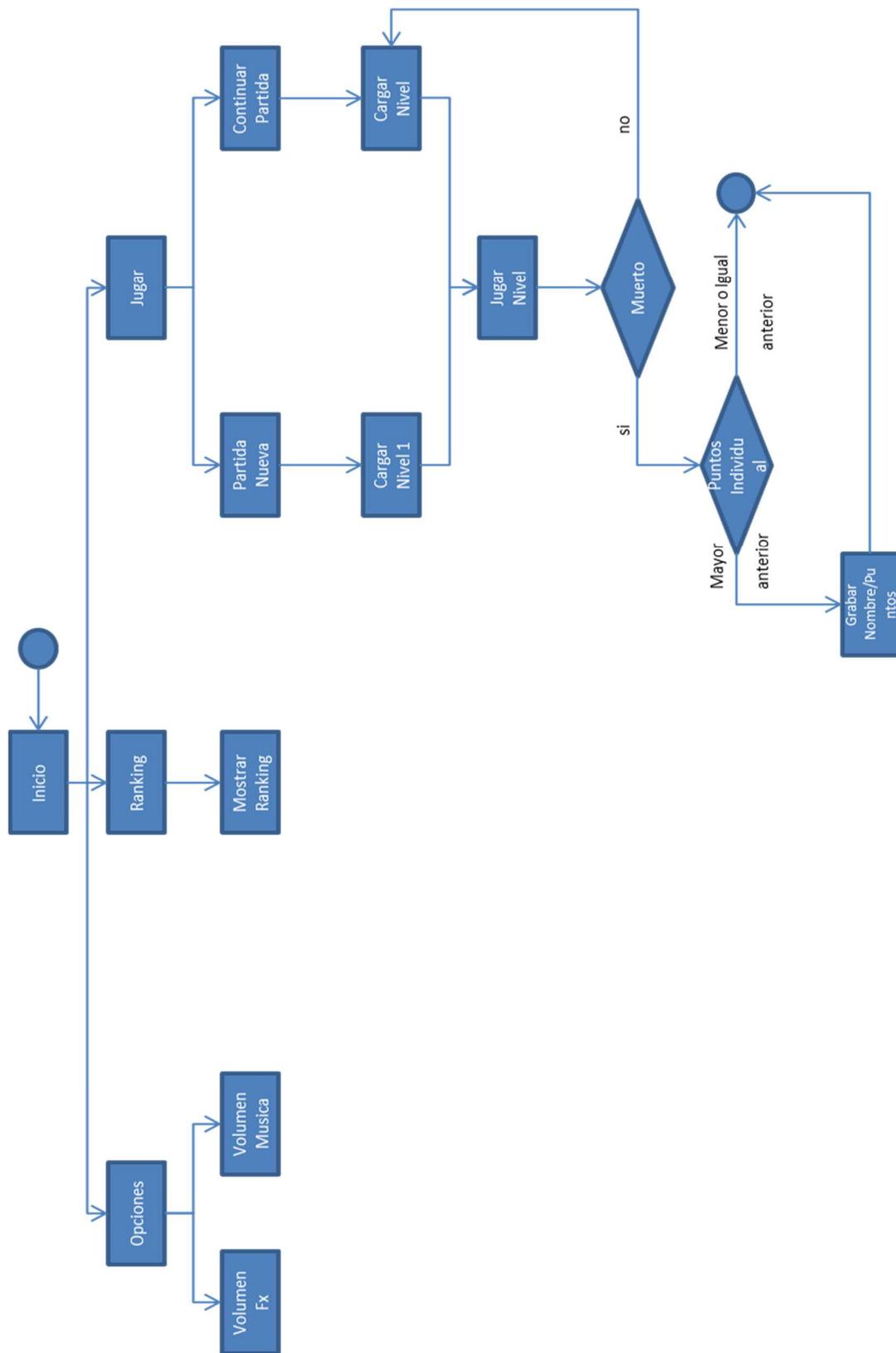


FIGURA 6 - DIAGRAMA ACTIVIDAD DEL SISTEMA

2.2 Diseño.

2.2.1 Prototipo.

2.2.1.1 Pantalla menú principal.

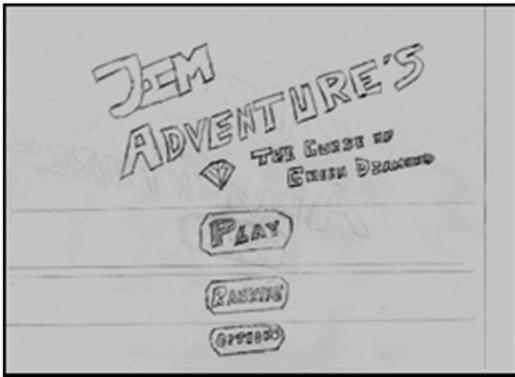


FIGURA 7 - SKETCH - PORTADA

En esta pantalla tenemos tres botones a los que el usuario puede acceder:

Play → Es botón para ir a jugar, que nos llevara a la pantalla que se muestra en Figura 6.

Ranking → Es el botón para ver el Ranking global (siempre que haya conexión con Internet) de los 10 mejores jugadores.

Options → Botón que nos trasladara a la sección de configuración del sonido.

2.2.1.2 Menú New Game o Continue (continuar una partida).



FIGURA 8 - SKETCH - NEW GAME

Esta pantalla se compone de dos botones:

- 1- New game
- 2- Continue

Si es la primera vez que se juega en este menú solo estará disponible el botón NEW GAME.

Para partidas posteriores y siempre que en la partida justamente anterior el jugador hubiera abandonado la partida sin terminarla y sin que le matasen se activaría el botón CONTINUE.

Pulsando este botón se continuaría la partida tal como estaba antes de abandonarla. En el caso de que el jugador decidiera pulsar NEW GAME se procedería a una partida totalmente nueva y se borrarían los datos de la partida a continuar y se iría a la pantalla Escenario 1.

2.2.1.3 Escenario 1.

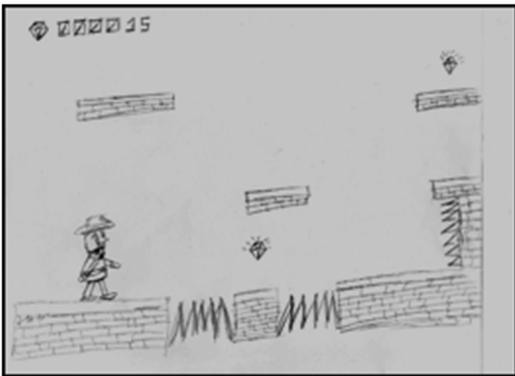


FIGURA 9 - SKETCH - ESCENARIO 1

Se trata de la pantalla principal del juego donde os encontramos con el personaje a la izquierda, que tendrá la habilidad de saltar entre plataformas o de rodar para evitar obstáculos.

En la zona superior izquierda dispondremos de un marcador de puntuación. El marcador se irá incrementando en puntos con respecto al tiempo que llevemos jugando e incrementándose también con respecto

a los objetos que recojamos.

Al comenzar cada nivel se mostrara el siguiente mensaje en pantalla.

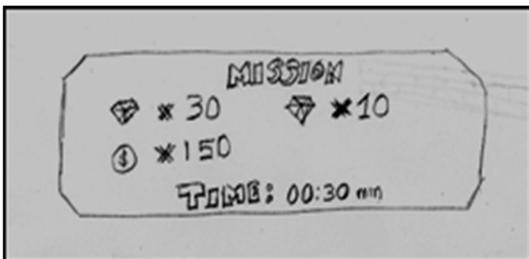


FIGURA 10 - SKETCH - PERGAMINO

En el tablón se nos mostrará los objetos que tenemos que coger para pasar de nivel y el tiempo disponible. El tablón desaparecerá a los tres segundos de cargar el escenario y ya comenzará la partida...

2.2.1.4 Escenario 2.

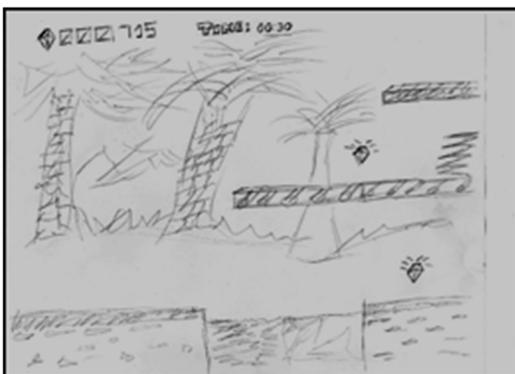


FIGURA 11 - SKETCH - ESCENARIO 2

Se trata del segundo escenario en el que saldría el personaje principal. Cada vez que se pasa de nivel se va rotando entre Escenario 1 y Escenario 2, cambiando las misiones. Se trata de una selva donde el personaje saltara de plataforma en plataforma.

2.2.1.5 Opciones.

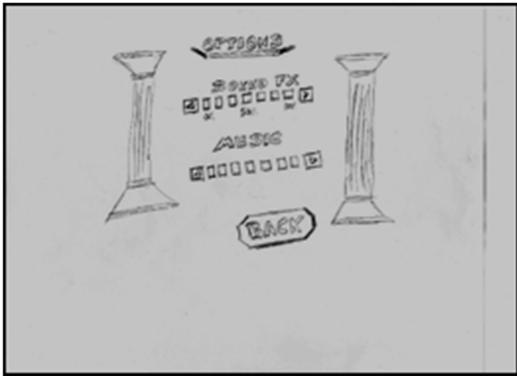


FIGURA 12 - SKETCH - OPTIONS

En esta pantalla se podrá cambiar el volumen tanto de los efectos especiales como de la música de fondo. Dependiendo si pulsamos el botón de la derecha o de la izquierda aumentará o disminuirá el volumen. Una barra de desplazamiento nos permitirá verificar que porcentaje de volumen vamos a aplicar a los distintos sonidos.

2.2.1.6 New Record.

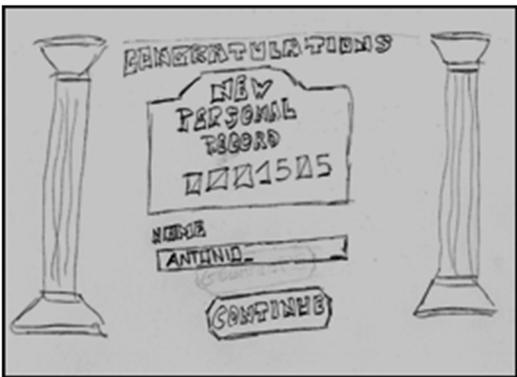


FIGURA 13 - SKETCH - NEW PERSONAL RECORD

A esta pantalla se accede una vez el personaje principal ha fallecido y se verifica que la puntuación obtenida es mayor que la que se tiene guardada de partidas anteriores.

Si es la primera vez que se juega también saldrá esta pantalla ya que si el jugador ha ganado aunque sea solo 1 punto habrá superado la puntuación básica, es decir, 0.

Una vez introducido el Nombre al pulsar el botón CONTINUE se procederá al grabado del récord.

Cuando se pulsa el botón CONTINUE puede pasar dos cosas:

- 1- Que haya acceso a Internet:

Una vez grabado en local se procederá al chequeo de conexión a Internet y si se tiene se comprobará si la puntuación sacada está entre las 10 mejores posiciones grabadas en Internet. Si es así, se grabará en la posición que corresponda borrando el último de la lista.

- 2- No haya acceso a Internet:

Se informará de ello y se volverá al menú principal.

2.2.1.7 Ranking.



FIGURA 14 - SKETCH - RANKING

Esta pantalla mostrará los 5 jugadores con mejor puntuación del juego. La información se recogerá siempre que haya conexión a Internet.

A esta pantalla se podrá acceder de dos maneras:

1- Cuando se haya conseguido un récord personal que supere a alguno de los que haya guardados en Internet.

2- Desde el menú principal, pulsando el botón RANKING.

2.2.2 Prototipo alta fidelidad.

A continuación se presentan los prototipos en alta fidelidad de cada una de las pantallas que se diseñaron como sketches.

2.2.2.1 Menú principal.



FIGURA 15 - PROTOTIPO - MENÚ PRINCIPAL

Este menú difiere en la posición de los botones con respecto al sketch para adaptarse mejor a la pantalla. Los botones se alinearon en horizontal en vez de en vertical como se imaginó en el boceto.

2.2.2.2 Menú New Game o Continue (continuar una partida).



FIGURA 16 - PROTOTIPO - MENÚ NEW GAME O CONTINUE

2.2.2.3 Escenario 1.

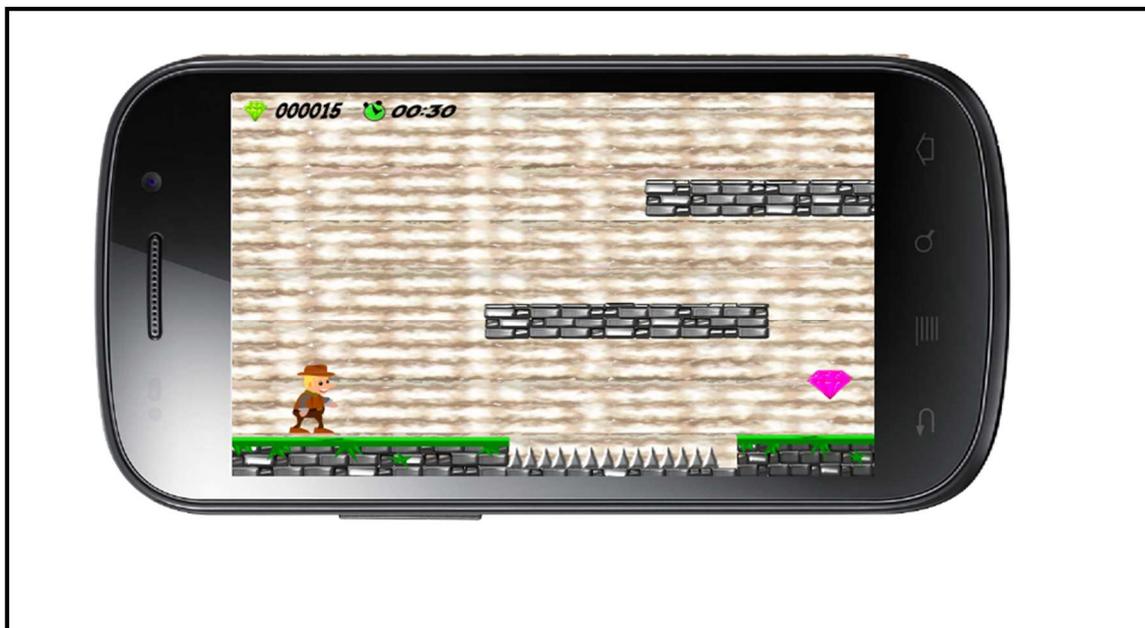


FIGURA 17 - PROTOTIPO - ESCENARIO 1

2.2.2.4 Escenario 2.



FIGURA 18 - PROTOTIPO - ESCENARIO 2

2.2.2.5 Opciones.



FIGURA 19 - PROTOTIPO - OPTIONS

2.2.2.6 New Record.

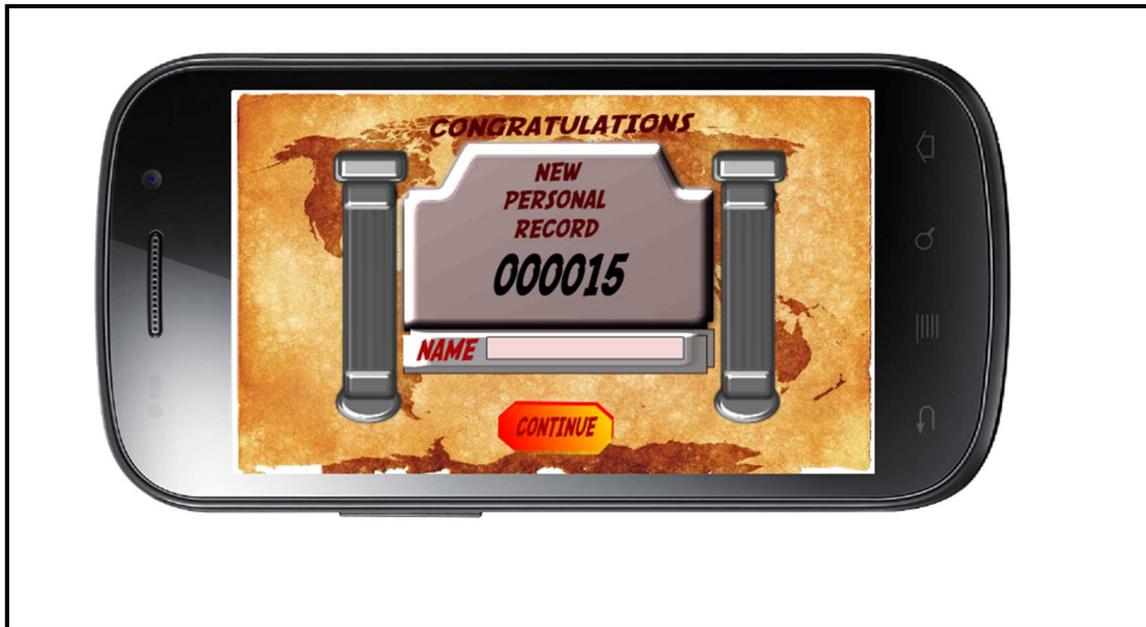


FIGURA 20 - PROTOTIPO - PERSONAL RECORDS

2.2.2.7 Ranking.



FIGURA 21 - PROTOTIPO - RANKING

2.2.2.8 Pergamino.

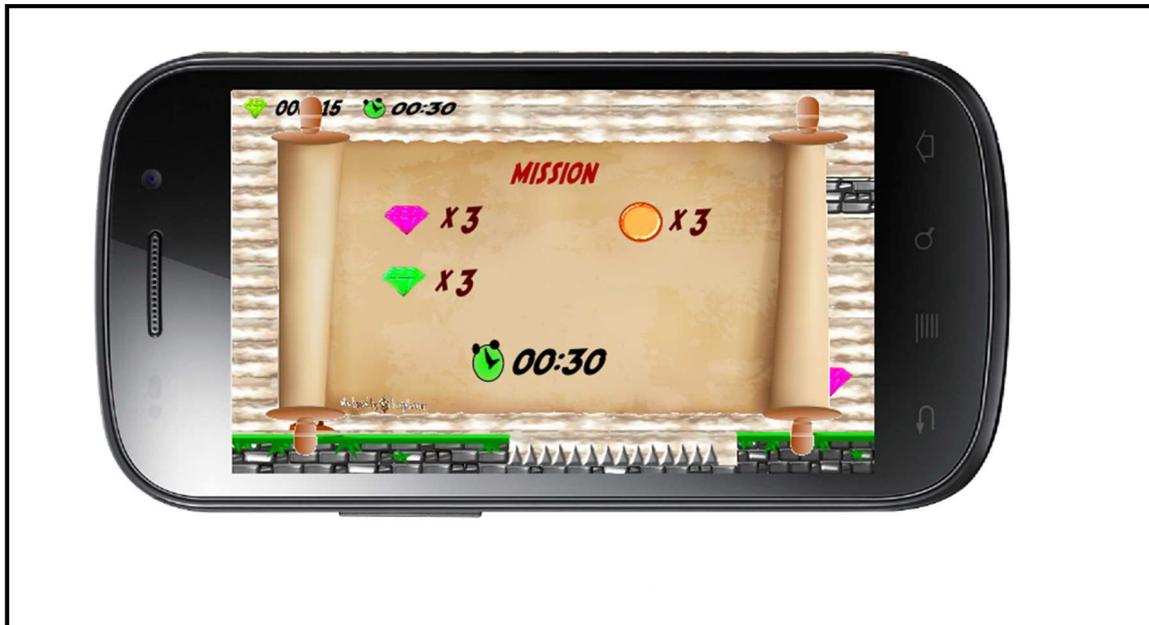


FIGURA 22 - PROTOTIPO - PERGAMINO

2.3 Implementación.

2.3.1 Objetivo.

El objetivo principal de este proyecto era la creación de un videojuego 2D de estilo Runner, para la plataforma Android®, utilizando Unity3D® como motor gráfico, crear Sprites e imágenes con InkScape® y utilizar Gimp® para la manipulación de estas.

La premisa era crear el videojuego para que sea fácilmente manejable en dispositivos móviles táctiles, cuyas misiones no duren mucho y sea divertido.

2.3.2 Escenas.

Unity® está formado por escenas. Normalmente cada escena correspondería a de las distintas pantallas del juego: inicio, niveles, opciones. En este caso para los niveles se optó por la creación de un módulo especial para la creación de niveles, grabarlos en ficheros y solo tener una escena donde se cargaran todos los niveles del juego.

A continuación se enumeraran las distintas escenas creadas en Unity©:



FIGURA 23 - PRODUCTO FINAL - PORTADA

1-**Portada:** Es la primera escena del juego. En esta escena tenemos los botones de Ranking, Play y Options. Al iniciarse esta pantalla se controla que todos los ficheros necesarios para el juego están creados y se pueden acceder a ellos sin ningún problema.



FIGURA 24 - PRODUCTO FINAL - ELECCIÓN

2-**Elección:** Esta escena nos permite elegir si empezar una partida nueva o seguir por el último nivel al que hemos jugado.

Botón New Game: Crea una partida nueva, es decir, empieza desde el nivel 1. Si el jugador está en el primer nivel solo aparece este botón, si está en un nivel superior aparecerán ambos.

En caso de estar en un nivel superior y pulsar este botón la partida guardada se reiniciara y por lo que se empezara en el nivel 1 de nuevo.

Botón Continue: Nos envía al nivel al que hubiéramos llegado con anterioridad, conservando la puntuación que hubiésemos conseguido.



FIGURA 25 - PRODUCTO FINAL - ESCENA 1

3-Escena 1: Es la escena que controla todos los niveles. Se encarga de cargar cada uno de los niveles creados, de controlar al personaje, de la destrucción de objetos y desplazamiento de estos por pantalla. También se encarga de mostrarnos las distintas metas a conseguir en cada nivel así como el pergamino de Game Over cuando el personaje ha muerto.



FIGURA 26 - PRODUCTO FINAL - NEW RECORD

4-New Record: Es la escena que nos muestra el nuevo record conseguido por el usuario. Se puede añadir el nombre de usuario que ha conseguido la nueva puntuación máxima y se comunica con Google Play para grabar la nueva puntuación si el usuario está conectado a internet y tiene una cuenta de Google Play.

En la última versión se ha quitado el poder introducir el nombre debido a que el GUI de Unity© no se adaptaba bien a las distintas pantallas. Además se vio que esta funcionalidad no aportaba nada desde el momento que se decidió implementar la API de Google play para guardar la información de las partidas en la red, ya que esta API usa el correo electrónico como identificador.



FIGURA 27 - PRODUCTO FINAL - LEVEL UP

5-Level Up: Esta escena se muestra cada vez que el usuario ha conseguido todos los objetos propuestos para pasar de nivel. Es la pantalla de transición entre los distintos niveles. Suena una música mientras nuestro personaje recorre el pasillo hasta llegar al margen superior derecho.



FIGURA 28 - PRODUCTO FINAL - OPCIONES

6-**Options:** Esta escena controla el volumen general de la música y los efectos especiales. Los botones situados a izquierda y derecha de cada barra permiten ajustar tanto el volumen de la música de fondo como el de los efectos especiales.

Cuando pulsamos el botón Return la configuración queda guardada en el fichero Setting.gd por lo que el usuario tendrá esa configuración la siguiente vez

que acceda a esta pantalla.



FIGURA 29 – PRODUCTO FINAL - CREDITOS

7-**Créditos:** Si el jugador consigue pasar los 18 niveles de que está compuesto Jim Adventure se le mostrara la pantalla de Créditos donde se nombra a los creadores de la música, texturas como la de mapeo y pergamino, etc.. además del nombre del programador y del diseñador gráfico.

2.3.3 Scripts.

Para la creación del juego se han tenido que generar Scripts en lenguaje C#. Estos se añaden a los objetos usando el Inspector. A continuación se explica cada uno de los Scripts generados y su uso.

- **AspectUtility.cs:**[5] Este script fuerza a la pantalla a tener el ratio deseado, y que no sea cambiado por la aplicación.
- **BotonContinueRecord.cs :** Este script está asociado al botón Continue de la pantalla New Record. Graba en el fichero Record el resultado de la puntuación y verifica si el usuario está conectado a Google Play, y si es así, añade su puntuación.

- **BotonContinueChoose.cs:** Este script está asociado al botón Continue de la pantalla de elección. Comprueba si el usuario está en el nivel 1 y si es así el botón continue no se hace visible. Da paso a la Escena1.
- **BotonFx.cs:** Controla el volumen de los efectos especiales. Va asociado a los botones Más y Menos volumen de efectos especiales de la pantalla de Opciones. Controla la barra de nivel que nos indica el porcentaje del volumen usado para FX.
- **BotonMusic.cs:** Controla el volumen de la música general. Va asociado a los botones Más y Menos volumen de música de la pantalla de Opciones. Controla la barra de nivel que nos indica el porcentaje del volumen usado para Música.
- **BotonNewGame.cs:** Controla el funcionamiento del botón New Game de la pantalla de elección. Graba en el fichero Status inicializando las variables nivel, puntuación, etc.. y da paso al primer nivel en Escena1.
- **BotonOptionsPrincipal.cs:** Controla el funcionamiento del botón Options de la pantalla principal. Recoge la información del fichero Settings sobre el volumen FX y Música y da paso a la pantalla Options.
- **BotonPlay.cs:** Controla el funcionamiento del botón Play del menú principal. Da paso a la pantalla de elección de juego nuevo o continuar por el nivel anterior.
- **BotonRanking.cs:** Controla el funcionamiento del botón Ranking del menú principal. Se conecta a google play para recoger la información del Ranking Global.
- **BotonReturnOptions.cs:** Controla el funcionamiento del botón Return del menú de opciones. Graba la información del volumen de efectos especiales y música en el fichero Setting.
- **CameraJim.cs:** Este scripts controla que la cámara siga al personaje principal. Da la sensación de que el personaje se está moviendo cuando en realidad es la cámara la que se desplaza.

- **CargaPergamino.cs:** Este script se encarga de visualizar el pergamino en pantalla y hacerlo desaparecer pasados 2 segundos. Carga los objetos y las cantidades de cada objeto para poder finalizar el nivel.
- **CargarNivel.cs:** Este script es uno de los más importantes. Se encarga de cargar la información relativa a cada nivel en la escena Escenario1. Construye con la información de los ficheros el nivel a ser jugado. Se accede a los ficheros Goals, Record, Status, Nivel y Setting para crear cada nivel.
- **ControlarFicheros.cs:** Este script se ejecuta al inicio del juego. Controla que todos los ficheros vitales para el juego estén creados. Sino intenta copiarlos desde la librería interna del juego (.apk) a las librerías de datos. Si algo falla muestra un mensaje con el error producido y se sale de la aplicación.
- **ControlJim.cs:** Esta clase controla los movimientos del personaje en pantalla. Si el usuario pulsa en la parte izquierda el protagonista salta, si el usuario pulsa en la mitad derecha de la pantalla el personaje rueda.
- **Creditos.cs:** Este script controla los nombres a ser mostrados en la pantalla de Creditos.
- **DetectorBucleNivel.cs:** Este script se encarga de mover los objetos del nivel ya que los niveles se basan en un bloque fijo de objetos que se van recolocando para dar la sensación de continuidad. Algunos bloques como los pinchos verticales solo están la primera vez y se destruyen en vez de ser movidos.
- **DetectorCaida.cs:** En los niveles avanzados hay objetos que son pasarelas móviles lo que deja el nivel 0 sin suelo por lo que si el personaje cae por ellas debe morir. Este script detecta si el personaje ha caído y si es así finaliza el juego.
- **DiamondDestructor.cs:** Este script se ha implementado para todos aquellos objetos que se quiere interactúen con el personaje principal, se contabilicen y se destruyan.

- **EntradaTexto.cs:** Se utiliza en la pantalla de New Record para introducir el nombre de la persona que ha alcanzado el record.
- **FondoScroll.cs:** Se utiliza para mover cíclicamente el fondo de pantalla. Se controla que el juego haya empezado antes de empezar a moverse.
- **Game.cs:** Estructura de datos usada para el fichero Nivel.
- **GameOver.cs:** Controla el fin de juego. Dependiendo de la puntuación del jugador decide si mostrar la pantalla de inicio o la pantalla con la puntuación New Record.
- **GeneradorNivel.cs:** Este script se utiliza para grabar en fichero Nivelxxx.gd los distintos niveles creados. Se trata de una utilidad para poder crear los niveles fácilmente, no forma parte del juego.
- **JimDetector.cs:** Este script permite la detección de colisión con el personaje principal. El objeto que lo lleve se convierte en un objeto dañino para Jim y por lo tanto lo matara.
- **JimPortada.cs:** Este script controla a Jim corriendo detrás del diamante verde en la portada principal.
- **LevelPergamino.cs:** Muestra el nivel que se va a jugar en el pergamino de inicio de nivel.
- **LevelUp.cs:** Controla al personaje corriendo y ascendiendo al siguiente nivel en la pantalla LevelUp. También controla la música que suena en esta transición.
- **Marcador.cs:** Controla los distintos marcadores del juego verificando si el personaje principal ha alcanzado los objetivos del nivel, y si se cumplen llamar al escenario LevelUp.
- **MostrarNewRecord.cs:** Muestra el nuevo record en la escena New Record.

- **NivelData.cs:** Estructura de datos que se utiliza para el fichero Goals. Contiene los datos de los objetivos a alcanzar en cada nivel
- **NotificacionCenter.cs:** Script usado para manejar mensajes entre diferentes GameObjects.
- **OcultarPergamino.cs:** Script usado para ocultar el pergamino que muestra las metas a alcanzar en cada nivel.
- **OptionsData.cs:** Estructura de datos que se utiliza para guardar en el fichero Setting la información relativa al volumen de la música y efectos especiales.
- **PersonalData.cs:** Estructura de datos que se utiliza para guardar en el fichero Status las monedas recolectadas y el nivel por el que el usuario va jugando.
- **PlataformaAndante.cs:** Script usado para elevar y bajar un objeto repetidamente.
- **RecordData.cs:** Estructura de datos que se utiliza para guardar en el fichero Record el nuevo record conseguido por el usuario.
- **VolumenFx.cs:** Script para cargar el volumen de los efectos especiales en los objetos.
- **VolumenMusic.cs:** Script para cargar el volumen de la música en los objetos.

2.3.4 Monobehaviour.

Es la clase base de la que derivan todos los scripts. Todos los scripts creados, menos aquellos de estructura de datos, tienen la clase Monobehaviour.

A continuación vamos a explicar las funciones principales de esta clase por orden de llamada:

- **Awake:** esta función es llamada antes que la función Start (comienzo) y solo cuando se carga la Escena.
- **Start:** es llamada antes de la primera actualización del frame.
- **FixedUpdate:** es llamado, a menudo, más frecuentemente que la función Update. Todos los cálculos del motor de física ocurren inmediatamente después de la función FixedUpdate.
- **Update:** esta función es llamada una vez por frame.
- **OnGui:** es llamada múltiples veces por frame en respuesta a eventos GUI.
- **OnDestroy:** esta función es llamada después de haber actualizado todos los frame de los objetos existente.

La imagen de la siguiente página nos muestra el orden de ejecución de las distintas funciones de la clase MonoBehaviour.

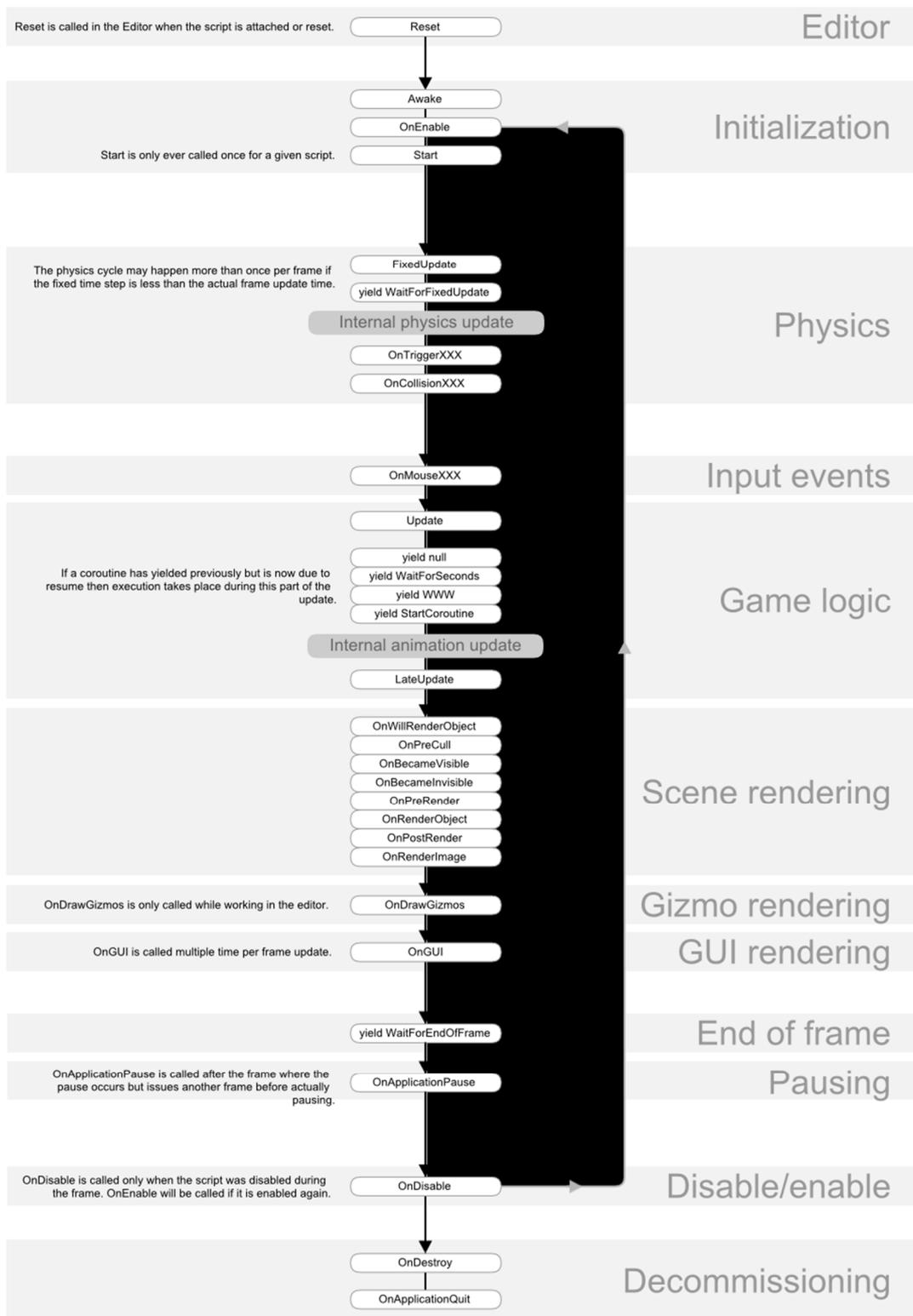


FIGURA 30 - MONOBEHAVIOUR - ORDEN EJECUCIÓN DE FUNCIONES

2.3.5 Ficheros.

Durante el juego se manejan un cierto número de ficheros que guardan la información de la partida, estado del juego, del volumen, niveles, etc....

- **Goalx.gd:** Contiene la información de la misión que hay que cumplir en cada nivel. X es el número de nivel del que contiene información.
- **Nivelx.gd:** Contiene la información de todos los objetos que conforman cada nivel a ser cargado. X es el número de nivel del que contiene información.
- **Setting.gd:** Contiene la información del volumen parametrizado para la música y para los efectos especiales.
- **Record.gd:** Contiene la información del record personal en este dispositivo.
- **Status.gd:** Contiene la información de la puntuación y el nivel por el que va el jugador.

2.3.6 Requisitos del Sistema.

Jim Adventure necesita la siguiente configuración mínima para poder ejecutarse:

- Android®: versión OS 2.3.1 (conocida como Gingerbread) o posterior.
- Procesador ARMv7 (Cortex) CPU o Atom CPU.
- OpenGL ES 2.0 o posterior.
- Tener instalado Google Play Services.

Funciona tanto para móviles como tabletas, siempre y cuando tenga la mínima configuración especificada anteriormente.

2.3.7 Pruebas.

Los usuarios de pruebas/tester han realizado las siguientes tareas para verificar el software.

TAREA PRINCIPAL	<i>Nueva partida</i>		
<i>Nombre tester</i>			
TAREA	OBSERVACIONES	RESULTADO ESPERADO	RESULTADO DEL TEST
Pulsar PLAY en el menú principal		Se muestra la pantalla NEW/CONTINUE donde el botón CONTINUE no aparece. Si se presiona NEW se pasara al juego en el nivel 1i.	
Pulsar en la parte derecha de la pantalla		Jim salta	
Pulsar en la parte izquierda de la pantalla		Jim rueda	
Coger objetos		El objeto desaparece, se incrementa puntos	
Saltar sobre pinchos o áreas donde el personaje tiene que morir		El juego se para y sale el letrero de GAME OVER	
Saltar entre distintas plataformas hasta cumplir el tiempo indicados sin recoger los objetos necesarios para superar el nivel		El juego se para y sale la pantalla de GAME OVER.	
Recoger todos los objetos antes del tiempo establecido para superar el nivel.		Se pasa al siguiente nivel y se incrementan los puntos correspondientes al tiempo que no hemos consumido. Se carga el siguiente escenario.	

TAREA PRINCIPAL		<i>Nueva partida</i>	
Nombre tester			
Superar la puntuación personal		Tras terminar la partida la pantalla sale la pantalla de récord personal.	
Superar la puntuación de Internet sin conexión a Internet		Saldrá un mensaje de que no hay conexión a Internet por lo que no se puede guardar la información de la partida. Se vuelve al menú principal	
Superar la puntuación de Internet con conexión a Internet superando la puntuación personal		Una vez insertado el nombre en la pantalla de récord personal saldrá la pantalla de Ranking con el nombre introducido y la nueva puntuación	
No se supera la puntuación personal pero si alguno del rango de Internet		Saldrá una pantalla donde introducir el nombre y tras hacerlo y continuar saldrá la pantalla de Ranking con el nombre introducido y la nueva puntuación	

TAREA PRINCIPAL		<i>Continuar partida.</i>	
Nombre tester			
TAREA	OBSERVACIONES	RESULTADO ESPERADO	RESULTADO DEL TEST
Pulsar el botón Play menú principal		Se muestra la pantalla NEW/CONTINUE donde el botón CONTINUE estar activo y al pulsar en CONTINUE se inicia la partida por donde la habíamos dejado.	
Verificar si el nivel cargado es el nivel que corresponde		Se carga el nivel correspondiente	

TAREA PRINCIPAL Nombre tester	Continuar partida.		
Coger objetos		El objeto desaparece, se incrementa puntos	
Saltar sobre pinchos o áreas donde el personaje tiene que morir		El juego se para y sale el letrero de GAME OVER	
Saltar entre distintas plataformas hasta cumplir el tiempo indicados sin recoger los objetos necesarios para superar el nivel		El juego se para y sale la pantalla de GAME OVER.	
Recoger todos los objetos antes del tiempo establecido para superar el nivel.		Se pasa al siguiente nivel y se incrementan los puntos correspondientes al tiempo que no hemos consumido. Se carga el siguiente escenario.	
Superar la puntuación personal		Tras terminar la partida la pantalla sale la pantalla de récord personal donde introduciremos el nombre.	
Superar la puntuación de Internet sin conexión a Internet		Saldrá un mensaje de que no hay conexión a Internet por lo que no se puede guardar la información de la partida. Se vuelve al menú principal	
Superar la puntuación de Internet con conexión a Internet superando la puntuación personal		Una vez insertado el nombre en la pantalla de récord personal saldrá la pantalla de Ranking con el nombre introducido y la nueva puntuación	
No se supera la puntuación personal pero si alguno del rango de Internet		Saldrá una pantalla donde introducir el nombre y tras hacerlo y continuar saldrá la pantalla de Ranking con el nombre introducido y la nueva puntuación	

TAREA PRINCIPAL		<i>Cambiar configuración de sonido</i>	
Nombre tester			
TAREA	OBSERVACIONES	RESULTADO ESPERADO	RESULTADO DEL TEST
Subir el volumen de la música		Cuando vuelve a jugar la música está más alta	
Subir el volumen de los efectos de sonido		Cuando vuelve a jugar el sonido FX está más alto. Al coger una gema, al saltar.. etc.	
Subir el volumen de la música		Cuando vuelve a jugar la música está más baja	
Subir el volumen de los efectos de sonido		Cuando vuelve a jugar el sonido FX está más bajo. Al coger una gema, al saltar.. etc.	
Pulsar el botón BACK		Se vuelve al menú principal	

TAREA PRINCIPAL		<i>Visualizar Ranking</i>	
Nombre tester			
TAREA	OBSERVACIONES	RESULTADO ESPERADO	RESULTADO DEL TEST
Visualizar Ranking sin acceso a Internet		Sale un mensaje diciendo que no hay conexión y que no se puede mostrar la lista	
Visualizar Ranking con acceso a Internet		Se muestra la pantalla con la lista de los 5 mejores jugadores con sus puntuaciones	

3. Introducción a los programas utilizados

Para la implementación de Jim Adventure se ha optado por el Unity3D®, una plataforma para la creación de juegos tanto 3D como 2D. El lenguaje usado ha sido C#, dentro de los tres lenguajes que se podían escoger (Javascript y Boo, eran las otras dos opciones).

La gran ventaja de Unity 3d® es que tiene herramientas en el entorno de programación que facilitan no solo la programación sino la utilización de objetos, física, animaciones, luces, etc..., necesarios para hacer un juego de calidad.

3.1 InkScape.

Se trata de una herramienta de diseño gráfico vectorial en formato SVG, gratuito, libre y multiplataforma. Las características de SVG soportadas incluyen formas básicas, trayectorias, texto, canal alfa, transformaciones, gradientes, edición de nodos, exportación de SVG a PNG, agrupación de elementos, etc.

Casi la totalidad de los gráficos usados en Jim Adventure han sido creados con esta herramienta.

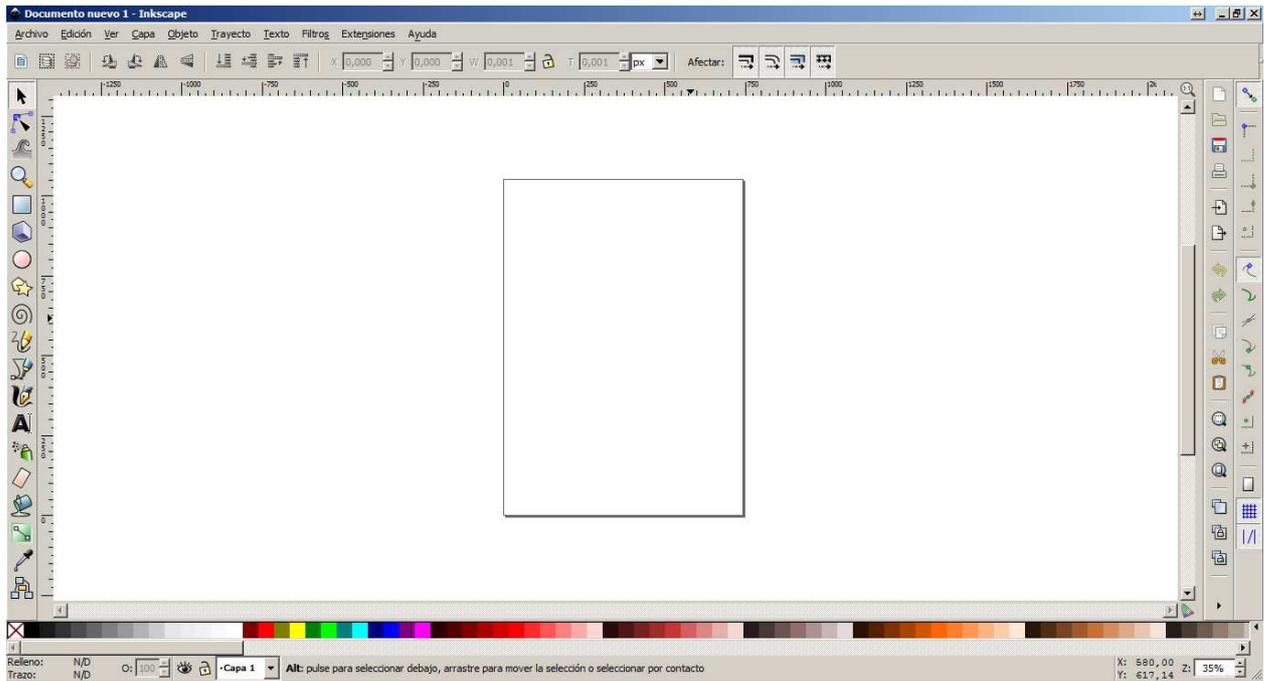


FIGURA 31 - INKSCAPE - INTERFAZ USUARIO

Comenzando con formas básicas, como círculos, cuadrados, etc.. y moldeándolos usando la herramienta de edición de nodos de trayecto se pueden crear formas complejas.

InkScape© tiene un diseño sencillo, objetos que se pueden usar para crear cualquier forma y herramientas para su moldeo están situados a la izquierda.

La parte central se utiliza como lienzo, mientras que en la parte inferior tenemos la paleta de colores y en la parte superior los menús desplegables con todo tipo de opciones.

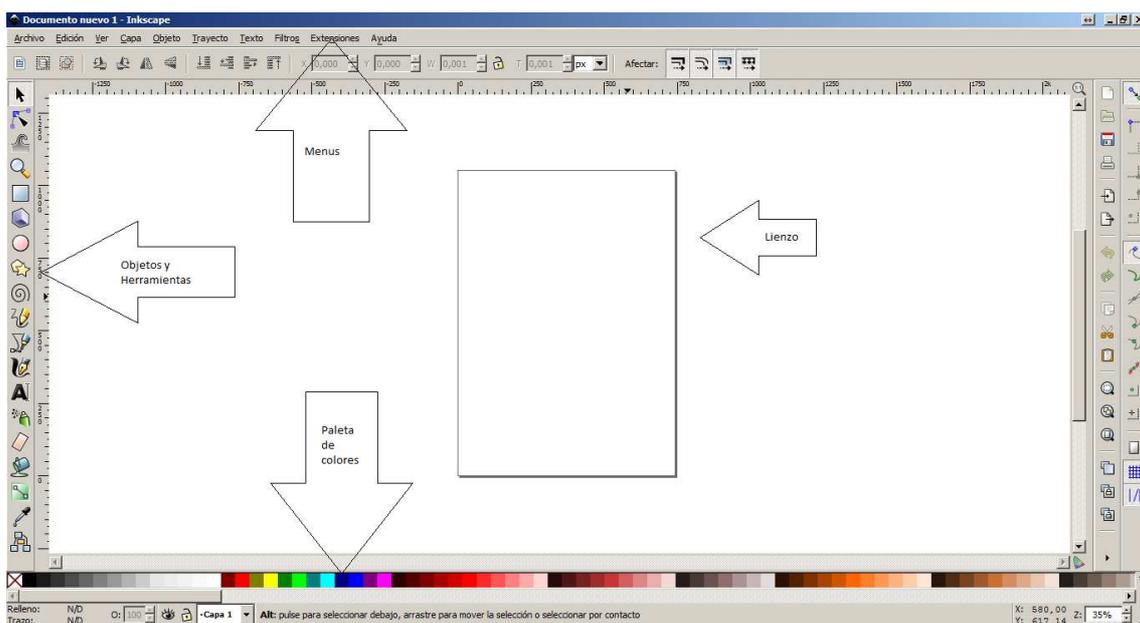


FIGURA 32 - INKSCAPE - BLOQUES

A continuación se explica cómo se ha usado este programa para crear el personaje principal, así como las plataformas y demás objetos usados en el juego.

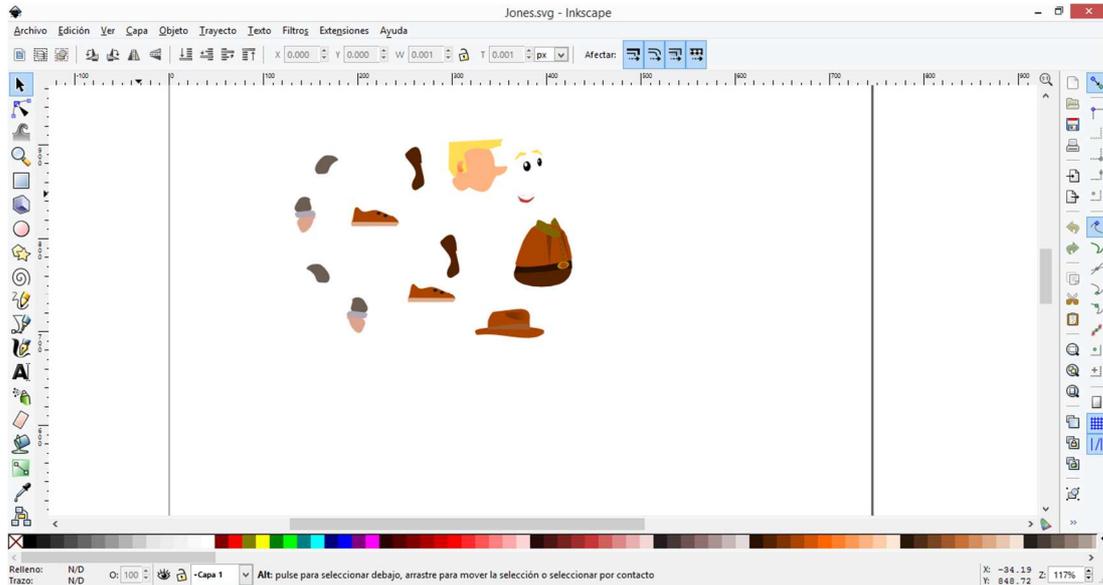
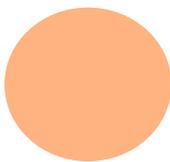


FIGURA 33 - INKSCAPE - LIENZO CON PERSONAJE

Las funciones más usadas se van a ilustrar a continuación con la creación del personaje principal.

En primer lugar para la elaboración de la cabeza generamos un círculo.



Nos vamos a opción de trazo y pinchamos en "Sin Relleno" así nuestro objeto no tendrá ningún trazo rodeándolo.

Convertimos el círculo en un path (trayectoria) y con la combinación de teclas **Mayus + Ctrl + C** (opción de menú Trayecto/Objeto a Trayecto) creamos la nariz.



Usando esta opción el círculo es totalmente moldeable por lo que se puede crear ya una cabeza como la mostrada.

Con este mismo procedimiento podemos a la creación de los ojos. Los ojos son dos círculos rellenos de color negro a los que incluimos dos círculos de menor tamaño de color blanco.

Otra funcionalidad muy usada es la de agrupación. Permite agrupar varios objetos para poder transformarlos simultáneamente, y no tener que hacerlo objeto a objeto. Los dos círculos que forman un ojo, por ejemplo,

deberían de ser un objeto único, para ello vamos a seleccionar los dos objetos pinchando sobre ellos y dejando presionado **Ctrl**. Una vez que tenemos los dos objetos que queremos unir nos vamos al menú Objeto/Agrupar (**Ctrl + G**) y a partir de ese momento esos dos objetos independientes que formaban el ojo pasan a ser uno, y podemos transformarlos como si fuese uno.

También tenemos que tener en consideración el orden Z de cada objeto. Si al juntar la cabeza con los ojos vemos que los ojos se ocultan tras la cabeza significa que la cabeza está en un plano o capa superior.

Con este mismo procedimiento podemos seguir hasta conseguir crear cada una de las partes del personaje principal, siempre usando rectángulos o círculos y modificando el path a nuestra conveniencia.

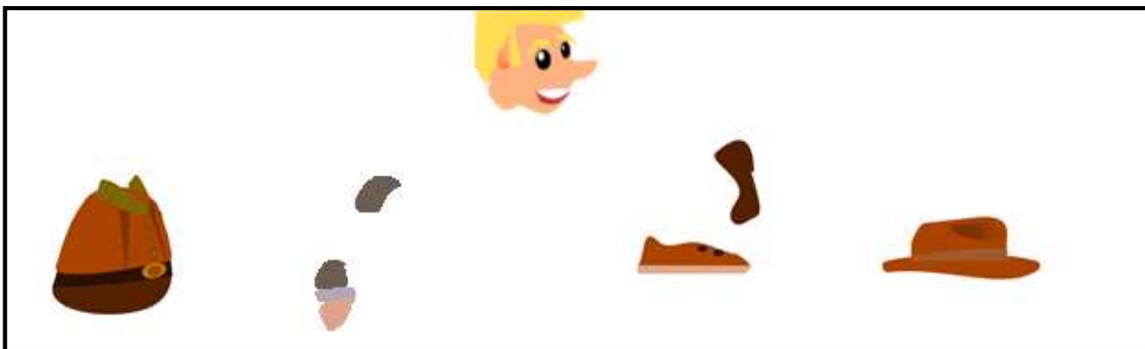


FIGURA 34 - INKSCAPE - PERSONAJE PRINCIPAL

Para las plataformas y fondos se ha usado el mismo procedimiento, pero esta vez jugando con los degradados y con efectos especiales predefinidos que vienen en InkScape®.



FIGURA 35 - INKSCAPE - PLATAFORMA

Con estas técnicas se ha conseguido hacer todo el diseño necesario para el juego a excepción del gráfico mapa de fondo usado en la pantalla del menú principal, opciones y elección, y el pergamino, usado para mostrar los objetivos de cada nivel, que se ha recogido de internet ya que tienen licencia Creative Commons.

3.2 Unity3D.

Unity3d© es un motor de videojuegos multiplataforma creado por Unity Technologies©.

Se dispone de una versión gratuita y una versión de pago, Unity© y Unity Pro© respectivamente, teniendo la primera, limitaciones en sus capacidades técnicas.

Nos permite trabajar para distintas plataformas con una sola herramienta. Así que con casi el mismo código o con pequeñas variaciones nuestro producto puede ser exportado a plataformas tan dispares como: Web, iOS©, Android©, Wii©, Xbox 360© y PlayStation 3©.

A partir de la versión 4.0 de Unity© se permite la creación de juegos 2D, ya que la primera finalidad de este motor fue la de generación de juegos en 3D.

3.2.1 Características.

Unity© se compone de:

- Editor: Entorno integrado de desarrollo, donde podemos disponer de jerarquías de objetos, visión de la escena y objetos que estamos manejando, ejecución del juego en entorno Unity©, etc.
- Assets: Se aceptan multitud de formatos ya sean en forma de Sprites, Texturas, u objetos 3D generados con herramientas tan populares como Blender, Maya, etc...
- Gráficos: Los gráficos están optimizados para Direct X y Open Gl. A partir de la versión 4 se incluye además de las funcionalidades de manejo de gráficos la de Animación con un sistema propio llamado Mecanim.
- Red: Se pueden acceder a funciones de Red para generar juegos multijugador, habilidades sociales, etc.
- Audio y Video: Provee de funciones de audio y video (esta última está bastante limitada en la versión gratuita), ya sea audio en 3d como en 2d.

- Scripting: Tiene tres lenguajes de programación con los que generar nuestro código, C#, javascript, y Boo. Además posee un editor de código propio llamado MonoDevelop.

3.2.2 Interfaz de usuario.

La primera vez que accedemos a Unity 3D© se nos pide si nuestro proyecto va a ser de tipo 2D, o 3D.

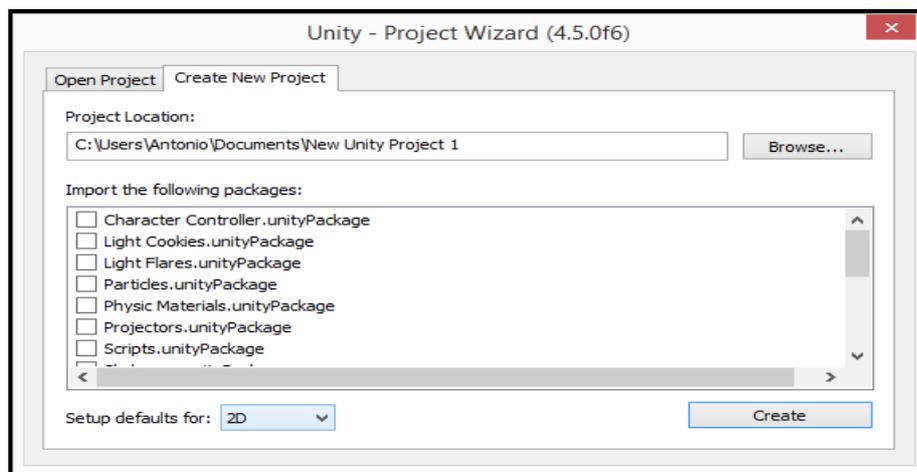


FIGURA 36 - UNITY 3D - PROJECT WIZARD

Una vez creado el proyecto se nos presenta una ventana como la siguiente:

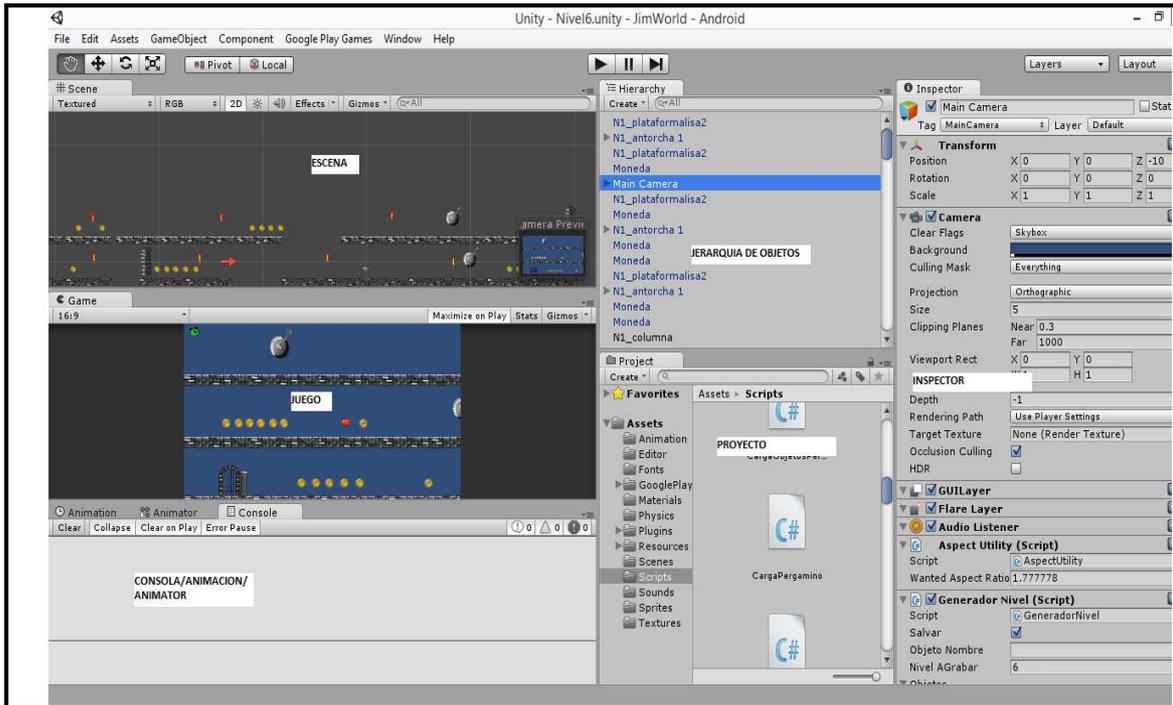


FIGURA 37 - UNITY3D - DISTRIBUCIÓN VENTANAS

Podemos observar que tiene varias ventanas bien definidas:

1- **Scene (Escena):** Se trata de la pantalla principal donde vamos a ir colocando los objetos o GameObject para la generación del juego. Es donde también tendremos uno de los principales objetos (la cámara).

Como podemos observar en la parte inferior, aunque nuestro proyecto sea 2D la visión de los objetos se puede hacer en 3D. Esto nos ayuda a posicionar objetos con más profundidad que otros.

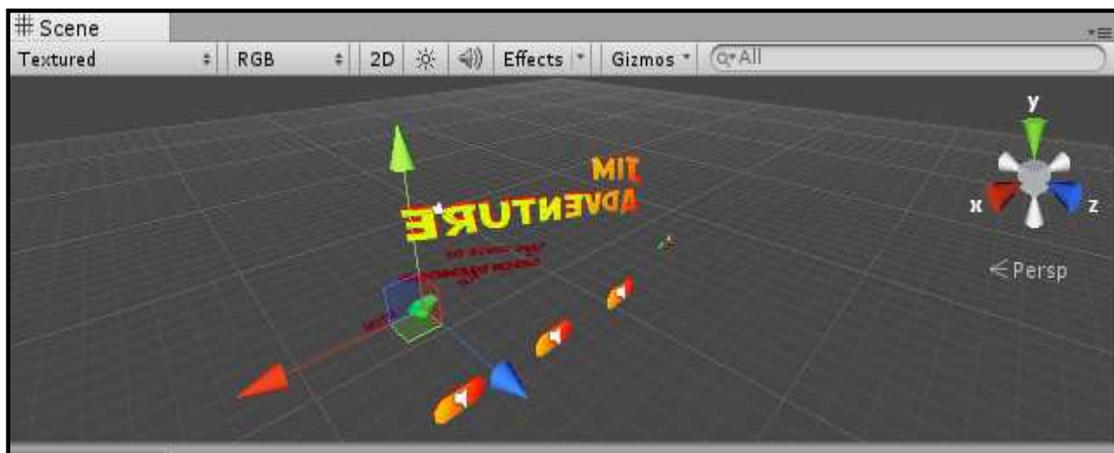


FIGURA 37 - UNITY3D - SCENE



En la parte superior izquierda de la pantalla, justo debajo de los menús desplegables, tenemos una serie de botones que nos permiten hacer diversas actividades en los objetos o la escena:

- Hand Tool: Este control nos permite movernos alrededor de la vista de la escena.
- Translate Tool: Nos permite mover cualquier objeto en ejes x, y, z.
- Rotate Tool: Nos permite rotar cualquier objeto de la escena.
- Scale Tool: Nos permite escalar cualquier objeto de la escena.

2- Hierarchy (Jerarquía de objetos): Cada vez que insertamos en nuestra escena un nuevo objeto o GameObject se crea una instancia en la ventana de Jerarquía de Objetos. Así podremos observar si ese objeto tiene algún tipo de jerarquía asociada, es decir, es padre o hijo de otro objeto o no.

3- Inspection: En esta ventana tendremos acceso a las propiedades públicas de cada objeto, Scripts asociados, etc.. Estas propiedades se

pueden cambiar sin necesidad de acceder al Script correspondiente o también, y muy útil, cambiarlas en tiempo de ejecución para hacer mejoras y ver su comportamiento. Permite también la inclusión o exclusión de características, scripts, etc..

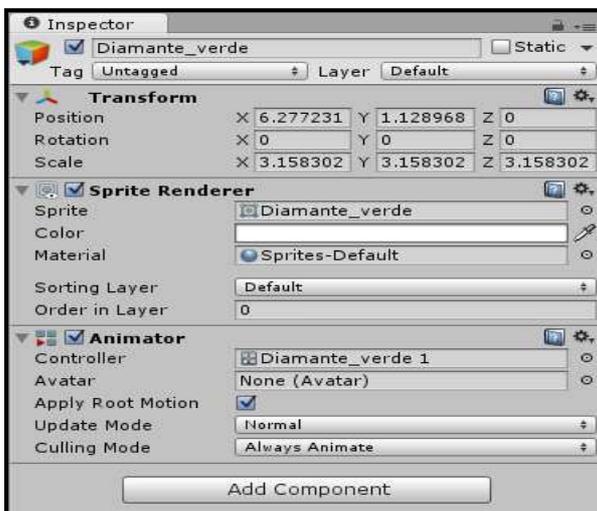


FIGURA 38 - UNITY 3D - VENTANA INSPECTOR

En la imagen podemos observar las propiedades del GameObject "Diamante Verde". La propiedad más importante del GameObject en Unity® es el método Transform. Con este método podemos definir su posición, rotación y escala.

4- **Project:** En esta ventana tenemos por un lado la jerarquía de carpetas que vamos a usar para guardar los distintos objetos, prefabs, texturas, música, etc.. Todas serán subcarpetas de la carpeta Assets.

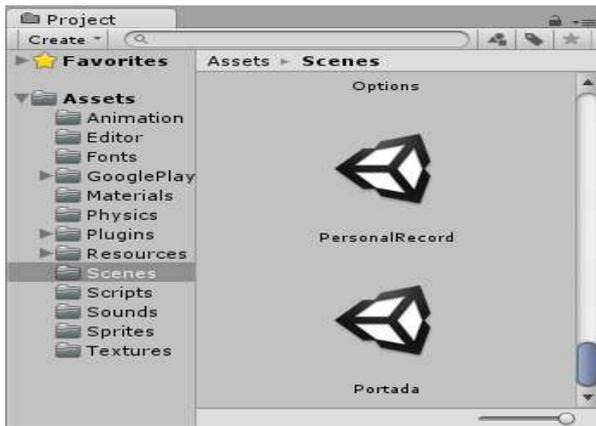


FIGURA 39 - UNITY3D - VENTANA PROJECT (PROYECTO)

5- **Animation/Animator/Console:** Estas tres pestañas contienen la información relativa a la animación de Sprites, a la creación de animaciones y a la consola del debugger de Unity© respectivamente.

6- **Game (Juego):** Es la vista que nos permite visualizar el juego en cualquier momento. Mientras que no hayamos pulsado el modo de reproducción nos muestra la imagen de la cámara principal en su inicio y todos los objetos que entren el objetivo de esa cámara.



Los botones mostrados anteriormente nos permiten entrar en pre visualización del juego, pausarlo en un momento determinado o saltar hacia adelante frame a frame.

Play – activa la pre visualización de la escena actual. Todos los controles, animaciones, sonidos etc..., empezarán a funcionar tal y como lo haría si el juego estuviera instalado ya en una terminal.

Pause – Pausa la ejecución de la escena que se está pre visualizando.

Step – Este botón solo se activa mientras el botón pause este activado. Se utiliza para ver el comportamiento del juego frame a frame.

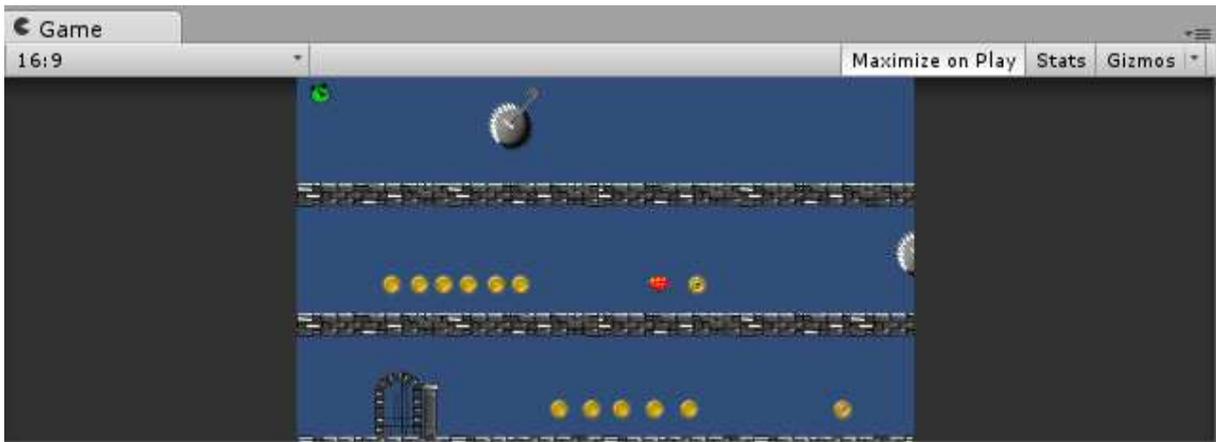


FIGURA 40 - UNITY3D - VENTANA GAME

Como podemos observar en la pestaña Game también poseemos algunos desplegable y botones:

Botón de Aspecto – Situado en el borde superior izquierdo. Podemos cambiar el aspecto con el que se mostrara en pantalla nuestro juego. En nuestro caso hemos elegido el 16:9.

Maximize on Play – Si este botón está activado, cuando iniciamos la pre visualización, la ventana del juego se maximizara.

Stats – Nos da estadísticas de rendering usadas en la escena mientras el juego está en marcha.

Gizmos - Los Gizmos son usados para permitir un debug visual o bien para colocar ayudas en la vista de escena.

Por ultimo mencionar como crear el ejecutable de nuestro juego. Para ello necesitamos acceder al menú File/Build Settings.



Las escenas que componen el proyecto deben de ser incluidas en "Scenes in Build". Aquí seleccionamos también la plataforma a la que queremos exportar nuestro juego e incluimos parámetros para la construcción del paquete final.

FIGURA 41 - UNITY3D - BUILD PROJECT

4. Conclusiones.

Las conclusiones que se han sacado de este trabajo son las siguientes:

- Se ha conseguido el objetivo de crear un juego 2D para la plataforma Android® usando herramientas como Unity 3D® e InkScape® desde cero, totalmente jugable y con las opciones sociales esperadas.
- La realización de un proyecto de esta envergadura, por un solo programador/diseñador etc... es una ardua tarea. Para realizar un producto de calidad de este tipo es necesaria la colaboración de un especialista en diseño gráfico, programadores, en definitiva, un equipo de al menos tres personas, así como profundizar más en las distintas opciones que ofrece Unity® para mejorar la interacción con el usuario final.
- Los comentarios por parte de los testers/usuarios final son fundamentales. Ellos son los que proponen mejoras, ven errores, etc. Gracias a ellos sé que el producto final tiene que ser pulido ya que es complejo, no en uso, sino para poder pasar los niveles para la mayoría de usuarios. Solo unos pocos han conseguido avanzar hasta niveles altos.
- La planificación se ha conseguido seguir, casi al completo, con algunas modificaciones, ya que siempre hay factores que no se pueden controlar: picos de trabajo que no permiten seguir el horario establecido, scripts que han tenido que ser reescritos y búsqueda de funcionalidad y que no se conseguía lo deseado, etc.
- La metodología seguida es perfecta para este tipo de proyectos ya que la retroalimentación de información con los usuarios es indispensable para las mejoras del producto final y el ir iterativamente cambiando el software antes de finalizar el proyecto hace que el producto final sea de calidad.
- Gracias a este trabajo he sabido valorar las aplicaciones gratuitas o de pago, el esfuerzo que conlleva la creación de estos productos, etc. También me ha abierto la ventana a crear productos para Android® en mi tiempo libre y ver otras opciones para mi carrera profesional.
- Se me ha mostrado los distintos pasos para la creación, diseño de juegos para Android® así como las distintas formas monetizar el producto final. De hecho gracias a este proyecto ya estoy dado de alta como programador en Google Developer para poder publicar mis trabajos en Play Store.
- Creo que dedicándole más tiempo al estudio de Unity® se pueden generar productos de mucha calidad, utilizando funcionalidades de los teléfonos como GPS, realidad aumentada, etc...

5. Cambios con respecto al proyecto inicial.

Ha habido cambios con respecto al análisis inicial de la aplicación.

- En primer lugar se decidió, por falta de tiempo y de diseño, un solo escenario, en vez de dos alternos tal como se especificó en un principio. El motivo es que al hacer el diseño se vio que no tenía sentido en la historia tener un nivel en el que el personaje estuviese en el castillo y el siguiente nivel en la selva. Se consideró tener bastante más niveles simplemente en el escenario 1 (se generaron 18 niveles) y dejar para una línea futura el implementar un escenario nuevo en el que el personaje se moviera en la selva.
- Otro cambio importante fue el sustituir el manejo del Ranking por la API de Google Play. Con esta API la información de las partidas está en internet por lo que da la sensación de más competitividad ya que el usuario puede ver sus resultados contra otros usuarios de distintos países. No hay que estar manejando los Ranking en la misma aplicación. Un efecto colateral de esta decisión ha sido que el usuario tiene que instalar Google Play para poder jugar al juego.
- La introducción del nombre de usuario en el Ranking se ha desechado debido a la implementación de Google Play, ya que esta usa el correo electrónico ligado a la cuenta del usuario para identificarlo. Además el GUI para introducir datos de Unity© tiene la problemática de que no se adapta bien a los tipos de pantalla por lo que en algunos dispositivos se veía desplazado.

6. Líneas Futuras.

- Debido a la limitación temporal del proyecto tampoco se ha podido incluir el tema de la publicidad como financiación del proyecto. Para ello sería necesario abrir una cuenta comercial en Google Play e implementar, en este caso, los interstitial ad APIs en el producto final.
- Se pretende generar hasta 50 niveles, más 5 niveles especiales que estarán ocultos, para la versión que será subida a Play Store.
- Se añadirán más funcionalidades sociales, como Medallas y se limitara el número de partidas que se puede jugar al día, pudiéndose comprar partidas extra mediante Play Store.
- En niveles superiores habrá enemigos que se tendrán que eliminar mediante acciones haciendo el juego más divertido y se dispondrá de un fondo más detallado.

5. Glosario

Android	Android© es un sistema operativo diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, perteneciente a la empresa Google.
Animation	Animación (traducción del inglés). Es el módulo de Unity3d© para la generación de animaciones en Sprites.
Animator	Animador (traducción del inglés). Es el modulo que controla las transacciones entre las distintas animaciones.
Arcade	El término "arcade" se refiere a los videojuegos clásicos o que recuerdan a las máquinas del mismo nombre.
BlackBerry OS	Sistema Operativo móvil desarrollado por RIM para los dispositivos Black Berry.
BlackBerry	Marca de teléfono inteligente desarrollada por la compañía canadiense BlackBerry.
Boo	Lenguaje de programación orientado a objetos.
C#	Lenguaje de programación orientado a objetos desarrollado por Microsoft.
Console	Se trata de un log donde se visualizan los errores, alertas y mensajes de la aplicación que se está desarrollando.
Debugger	Aplicación o herramienta que permite la ejecución controlada de un programa o código.
Gantt	Un diagrama de Gantt es una representación gráfica y simultánea tanto de planificación como de programación concreta de procesos y/o proyecto desarrollada por Henry L. Gantt
Gimp 2.0	Es un programa de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías. Es un programa libre y gratuito.
GUI	Interfaz Gráfica de Usuario. Conjunto de formas y métodos que posibilitan la interacción de un sistema con los usuarios utilizando formas gráficas e imágenes. Con formas gráficas se refiere a botones, íconos, ventanas, fuentes, etc. los cuales representan funciones, acciones e información.
InkScape	Editor de gráficos vectoriales en formato SVG, gratuito, libre y multiplataforma.
iOS	Sistema Operativo móvil desarrollado por Apple.
Javascript	Lenguaje interpretado de objetos.

Path	Rutas
Pause	Pausar (traducción del inglés). En Unity3D© se utiliza para parar los procesos que están corriendo en tiempo de ejecución en un momento determinado.
Plataforma	Son un género de videojuegos que se caracterizan por tener que caminar, correr, saltar o escalar sobre una serie de plataformas y acantilados, con enemigos, mientras se recogen objetos para poder completar el juego.
Play	Jugar (traducción del inglés). En Unity3D© se utiliza para comenzar la ejecución de la escena.
PNG	Portable Network Graphics. Formato gráfico.
Prototipo	Ejemplar original o primer molde en que se fabrica una figura u otra cosa.
Ranking	Clasificación de mayor a menor, útil para establecer criterios de valoración
Rendering	Proceso de generar una imagen a partir de un modelo.
Runners	Corredores (traducción del inglés). Se trata de un subgénero de los videojuegos de plataformas que se caracteriza porque el personaje principal está corriendo continuamente.
Scrum	Modelo de desarrollo ágil caracterizado por adoptar una estrategia de desarrollo incremental.
Step	Paso a paso (traducción del inglés). En Unity3d© es usado para ir frame a frame en tiempo de ejecución.
SUM	Modelo de desarrollo ágil que adapta para videojuegos la estructura y roles de Scrum.
SVG	Scalable Vector Graphics. Especificación para describir gráficos vectoriales bidimensionales.
Unity3d	Es un motor gráfico 3D para PC y Mac que viene empaquetado como una herramienta para crear juegos, aplicaciones interactivas.
Videojuego	Se define como Videojuego a toda aplicación o Software que ha sido creado con el fin del entretenimiento.
Windows Phone	Sistema Operativo móvil desarrollado por Microsoft.

6. Bibliografía

[1] **IDC (International Data Corporation)** - 25/09/2014 - <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

[2] **IDC (International Data Corporation)** - 25/09/2014 - <http://www.idc.com/prodserv/smartphone-market-share.jsp>

[3] **Jason Brownlee** - Mobile Games Engines - 22/09/2014 - http://www.mobilegameengines.com/android/game_engines

[4] **Varios autores** - Facultad de Ingeniería Universidad de la República, Uruguay - 25/09/2014 -

http://www.fing.edu.uy/inco/grupos/gris/wiki/uploads/Proceedings/ASSE_2009_16.pdf

[5] **Eric Haines** - AspectRatioEnforcer - 22/12/2014 -

<http://wiki.unity3d.com/index.php?title=AspectRatioEnforcer>

Juan de Dios Maldonado - Tutorial Creando un juego "Infinite Runner" en 2D con Unity 4.3 - 2/10/2014 - <https://www.youtube.com/watch?v=eBkdsNGI5ZQ&index=2&list=PLREdURb87ks2qkD9svllwYwN35FZ3Afv>

Unity Technologies - Manual de Referencia - 12/10/2014 -

<http://docs.unity3d.com/Manual/index.html>

Mike Geig - Unity Game Development in 24 hours - Pearson Education Inc. - Kindle Edition.

Gabriel Nieto - Unity 4x©. Tutorial de iniciación a la programación de videojuegos - Kindle Edition.

Andre Infante - Programming a game with Unity. A beginner's guide. – Kindle Edition

Dave Calabrese - Unity 2D game development – Kindle Edition

Varios autores – InkScape© Tutoriales – 2/10/2014 –

<https://inkscape.org/es/aprende/tutoriales/>

Chris Hildenbrand – 2D Game Art for Programmers –

<http://2dgameartforprogrammers.blogspot.com.es>

Varios autores – Música gratuita –

<http://incompetech.com/>

7. Anexos

Anexo I – Usuarios y contexto de uso:



Anexo I - Usuarios y
contexto de uso.docx

