**RESUMEN**

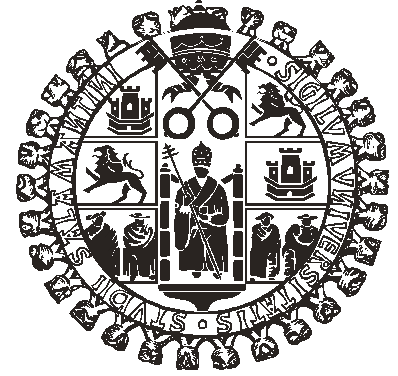
**TRABAJO DE FIN DE GRADO**

UNIVERSIDAD DE SALAMANCA

ESCUELA POLITÉCNICA SUPERIOR DE ZAMORA

GRADO EN INGENIERÍA INFORMÁTICA EN SISTEMAS DE INFORMACIÓN

**VIDEOJUEGO COOPERATIVO DE PLATAFORMAS EN 2D**



AUTOR: JORGE APELLÁNIZ COLINO

TUTOR: JUAN CARLOS MATOS FRANCO

*Departamento de Informática y Automática*

Fecha de adjudicación: Marzo 2019

Fecha de presentación: Septiembre de 2019

ÍNDICE

[1. INTRODUCCIÓN 3](#_Toc17723246)

[1.1. MOTIVACIÓN 3](#_Toc17723247)

[2. OBJETIVOS DEL PROYECTO 3](#_Toc17723248)

[3. CONCEPTOS TEÓRICOS 4](#_Toc17723249)

[4. HERRAMIENTAS 4](#_Toc17723250)

[4.1. MOTOR GRÁFICO UNITY 4](#_Toc17723251)

[4.2. UNITY MULTIPLAYER 4](#_Toc17723252)

[4.3. C# 5](#_Toc17723253)

[4.4. GITHUB 5](#_Toc17723254)

[4.5. FREE PLATFORM GAME ASSETS 5](#_Toc17723255)

[5. ASPECTOS RELEVANTES DEL DESARROLLO 5](#_Toc17723256)

[5.2. PLANTEAMIENTO DEL PROYECTO 5](#_Toc17723257)

[5.3. IMPLANTACIÓN 6](#_Toc17723258)

[6. PROBLEMAS ENCONTRADOS 7](#_Toc17723259)

[6.1. DESCONOCIMIENTO DEL MOTOR DE JUEGO Y EL LENGUAJE DE PROGRAMACIÓN 7](#_Toc17723260)

[6.2. RETARDO O *LAG* EN EL JUEGO MULTIJUGADOR 7](#_Toc17723261)

[6.3. PROBLEMA CON LAS ANIMACIONES EN RED 8](#_Toc17723262)

[6.4. INICIALIZACIÓN DE OBJECTOS DE TIPO COLLIDER AL INICIO 8](#_Toc17723263)

[7. CONCLUSIONES 8](#_Toc17723264)

[8. REFERENCIAS 10](#_Toc17723265)

# INTRODUCCIÓN

En este documento se busca reflejar la documentación relativa al Trabajo de Fin de Grado “**Juego cooperativo de plataformas en 2D**” relativo al Grado en Ingeniería Informática en Sistemas de Información de la Escuela Politécnica Superior de Zamora (Universidad de Salamanca).

Mediante el desarrollo y la implementación de esta aplicación informática se busca crear un videojuego, enfocado para ordenadores, basado en los juegos de plataformas, donde la mecánica de movimiento se puede asimilar a juegos conocidos como es *Mario Bros*. Se busca una experiencia de juego rápida y dinámica enfocada a ser disfrutada en pequeños tiempos muertos.

El objetivo es proporcionar un entretenimiento a corto plazo. El público objetivo serían personas con una edad superior a 8 años, A pesar de esto se espera que cualquier tipo de usuario pueda disfrutar de una experiencia de juego satisfactoria.

## MOTIVACIÓN

Desde principios de los años 80 la industria de los videojuegos ha sufrido un crecimiento enorme, superándose año tras año. Hasta el punto de convertirse en la mayor industria del entretenimiento a nivel mundial, superando a otros mercados del entretenimiento con creces.

Desde pequeño siempre me han gustado los videojuegos y por ese motivo y viendo que puede ser una salida profesional en este momento he decido desarrollar un videojuego con el que no solo sea aprender de esta rama de conocimiento sino también con poder disfrutar jugando a ellos.

# OBJETIVOS DEL PROYECTO

El objetivo principal de este Trabajo Fin de Grado es realizar una aplicación acompañada de una memoria explicativa en la que se pueda ver reflejada la aplicación de los conocimientos teóricos-prácticos adquiridos a lo largo de la etapa formativa, la capacidad de autosuficiencia e innovación mediante el trabajo con nuevas tecnologías.

Para la justificación de estas competencias se realiza un videojuego en 2D con una herramienta llamada “Unity3D”, construyendo la estructura principal de dicha plataforma, los componentes el diseño y las características.

El objetivo principal es la creación de un entretenimiento para un amplio rango de personas y poder satisfacerlas.

# CONCEPTOS TEÓRICOS

Para el desarrollo del videojuego se han seguido unos conceptos teóricos relacionados con el funcionamiento del motor gráfico Unity, más concretamente con su ciclo de vida, es decir, las fases que realiza el motor en el funcionamiento de un videojuego desarrollado con esta herramienta.

Todos los objetos que hay en el proyecto de Unity pueden llevar o no asociados una serie de Scripts donde se describe el funcionamiento propio de esos objetos.

Además, se han utilizado diagramas de estados para gestionar las animaciones de personajes.

# HERRAMIENTAS

Las herramientas utilizadas en este proyecto son las siguientes:

## MOTOR GRÁFICO UNITY

Unity es un motor de juego multi-plataforma desarrollado por la empresa *Unity Technologies.* Unity proporciona a los usaurios la posibilidad de crear juegos en 3D y 2D, para ello ofrece una API primaria de *scripting* en el lenguaje C#, tanto para el Editor de Unity en forma de *plugins* como para los juegos, así como la funcionalidad de arrastrar y soltar los elementos en la escena (*Drag&Drop*).

## UNITY MULTIPLAYER

Al implementar la opción de multijugador, Unity trae una herramienta llamada *UNET* (*Unity* *Multiplayer*), la cual hizo posible dicha implementación.

*Unity* *Multiplayer* es una de las mejores formas para crear juegos multijugador de forma rápida y sencilla. Dicho proyecto está preparado para que los jugadores se conecten mediante una conexión local. Pero Unity proporciona una serie de servidores (*Unity Matchmaker Servers*) que garantizan que los jugadores puedan encontrarse y jugar entre sí. [1]

## C#

C Sharp es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ISO (ISO/IEC 23270) y ECMA (ECMA-334).

Su sintaxis básica deriva de los lenguajes C/C++ y utiliza el modelo de objetos de la plataforma .NET, que es parecido al de Java, pero incluyendo una serie de mejoras derivadas de diversos lenguajes.[2]

## GITHUB

GitHub es un repositorio donde he podido ir almacenando los cambios que se han ido haciendo según se avanzaba con el desarrollo del proyecto. Cuando se hacían cambios importantes en el proyecto realizaba un commit con los cambios.

## FREE PLATFORM GAME ASSETS

Para los *sprites* del proyecto he utilizado un *asset* que proporciona Unity de forma gratuita, donde se encuentran diversos gráficos de enemigos, plataformas, personajes, objetos, etc. [3]

# ASPECTOS RELEVANTES DEL DESARROLLO

En este punto se van a destacar los aspectos más importantes para el desarrollo del proyecto desde la perspectiva del diseño e implementación.

Se ha utilizado una metodología de ciclo de vida de software específica. El modelo que se a utilizado para la realización del proyecto es el modelo iterativo incremental, ya que se permite crear versiones completas de software en cada iteración. Se selecciona este modelo ya que los requisitos no están todos establecidos inicialmente, y se pueden ir cambiando y añadiendo funcionalidades a la vez que se dispone de una versión funcional mejorable del producto.

## 5.2. PLANTEAMIENTO DEL PROYECTO

Desde un principio supe que quería realizar como trabajo de fin de grado un videojuego ya que me gustó bastante la asignatura de “Videojuegos” del grado y me pareció muy interesante. Se valoraron diferentes opciones de qué tipo de juego realizar: juego por turnos, *RPG*, etc. Pero finalmente la idea que me convenció fue un juego de plataformas que para añadir complejidad a la idea si impuso el tema de la cooperación entre 2 jugadores para lograr el objetivo del juego.

Una vez escogida la idea que quería desarrollar se lo comenté a mi tutor Juan Carlos Matos Franco, le pregunté su opinión sobre la viabilidad y complejidad del proyecto que me dio el visto bueno y accedió a tutorizar mi proyecto.

## 5.3. IMPLANTACIÓN

El proyecto se ha estructurado de tal forma donde podemos identificar 6 scripts como los más importantes: ‘LogicaJuego’, ‘MenuControl’, ‘menu’, ‘movimiento’, ‘temporizador’ y ‘persigueJugador’ (en el cual se implementa una simple inteligencia artificial). Éstos se explicarán a continuación de forma más detallada.

Por otro lado, tenemos el resto, los cuales contienen las funcionalidades propias de cada objeto las cuales se harán referencia a través de llamadas o instancias desde otros scripts.

* *LogicaJuego:* Script encargado de controlar todas las lógicas del juego.
* *Menu*: Script donde se encuentran los métodos para cambiar entre escenas.
* *Temporizador:* Script muy importante en el desarrollo del juego ya que en el se controla el tiempo de juego en la escena. Es decir, el tiempo que tiene el jugador o jugadores en caso de multijugador en lograr el objetivo.
* *Movimiento*: Script que controla el movimiento del personaje, la opción de disparo del jugador, entre otras.
* *MenuControl*: Script donde se controla el inicio del juego multijugador. En él se encuentran métodos para la conexión de la partida.
* *persigueJugador*: Script donde se implementa una simple inteligencia artificial aplicada a uno de los enemigos.

Para hacerse una pequeña idea de como es la escena del juego en sí, se puede ver en la Figura 1:



Figura . Escena de juego

# PROBLEMAS ENCONTRADOS

## DESCONOCIMIENTO DEL MOTOR DE JUEGO Y EL LENGUAJE DE PROGRAMACIÓN

Para poder hacer frente a este proyecto y el volumen de trabajo que conlleva ha sido necesario familiarizarse con el entorno de desarrollo de Unity3D, así como con el lenguaje de programación C#. Para estos objetivos se han realizado múltiples búsquedas en foros páginas web.

Uno de los puntos más favorables es la gran comunidad que está detrás de este motor gráfico, donde la gente está muy dispuesta a ayuda, además al llevar unos cuantos años funcionando muchas de las dudas que pueden surgir ya están resueltas en foros.

Por último, el lenguaje C# no es de una sintaxis muy compleja. Al ser orientado a objetos la forma de trabajo es parecida a Java, del cual se tenían algunos conocimientos previos, lo cual ha ayudado en gran medida a la hora de entender algunas de las funciones.

## RETARDO O *LAG* EN EL JUEGO MULTIJUGADOR

Uno de los problemas principales es el *lag* que surge entre el cliente y el servidor cuando están jugando en cooperativo. Esta herramienta que se ha utilizado que proporciona Unity (*UNET*) tiene el siguiente problema: cuando un jugador se mueve en su escena a su vez el otro jugador ve en su pantalla el movimiento de su compañero y ahí es donde se puede apreciar un pequeño retardo del personaje.

Esta tasa de retardo se puede reducir a lo mínimo incrementando este parámetro en el componente *NetworkTransform* del *prefab* del jugador:



Figura . Tasa de envío de red

Esta *tasa de envío de red* solo se puede configurar hasta ese valor (límite que proporciona la herramienta UNET). De esta manera se consigue disminuir bastante ese retardo entre jugadores. A pesar de ello siempre habrá un ligero retardo.

## PROBLEMA CON LAS ANIMACIONES EN RED

Debido a que la información de la herramienta *UNET* carece de mucha documentación en un primer momento al conectarse ambos jugadores a la partida no se conseguía enviar la información de las animaciones entre cliente y servidor.

Dando muchas vueltas y investigando a fondo se averiguó que existía un componente *NetworkAnimator* que proporcionaba esa opción.

## INICIALIZACIÓN DE OBJECTOS DE TIPO COLLIDER AL INICIO

En un primer momento se quería *instanciar* objetos de juego en la escena al inicio de ella, pero surgió el problema de que si algunos objetos chocaban con otros teniendo el componente “*Box Collider 2D”* las físicas no se habían iniciado y por tanto estos objetos no interactuaban. Como se puede ver en el ciclo de vida de Unity el método *Start* es anterior a las físicas.

La solución que se le dio finalmente ha sido colocar los objetos previamente en la escena de tal manera que no haya que instanciarlos en el método *Start*. De modo que se utilizar habilitándolos y deshabilitándolos dependiendo de un momento del juego u otro.

# CONCLUSIONES

La materialización de este proyecto ha supuesto un esfuerzo diario los últimos meses, al mismo tiempo ha valido para entender, aprender y sintetizar las fases que hay que realizar para llevar a cabo una idea que inicialmente se plasma en papel hasta que finalmente se escribo el su código construyendo una aplicación ejecutable.

Este proyecto es la culminación de todos los conocimientos, tanto teóricos como prácticos, adquiridos los últimos cuatro años de estudio en el Grado en Ingeniería Informática en Sistemas de Información. Gracias a la realización de este proyecto se han obtenido una gran cantidad de nuevos conocimientos los cuales serán de gran utilidad a la hora de enfrentarse a la inserción del mercado laboral.

Los aspectos técnicos logrados tras la realización del proyecto son los siguientes:

* Profundización de los conocimientos del motor gráfico de Unity, así como del lenguaje de programación C#.
* Ampliación de los conocimientos en cómo está situado la industria del videojuego tanto en España como en el mundo.
* Adquisición de nociones básicas en diseño de videojuegos.
* Utilización y comprensión de manera práctica del ciclo de vida del software, así como planificación a largo plazo de objetivos y posibles soluciones a los problemas acontecidos.

En cuanto a las conclusiones a nivel personal que se pueden sacar de la realización de este proyecto son las siguientes:

* Aprendizaje de la estructuración de proyectos con más volumen de trabajo y esquemas de programación.
* Oportunidad de emplear los conocimientos adquiridos a el mundo laboral.
* Capacidad de esfuerzo y aprendizaje autodidacta. Es la primera vez que se realiza un proyecto de esta envergadura por lo que se ha tenido que aprender a gestionar grandes volúmenes de trabajo y resolver los problemas que han ido apareciendo de forma autónoma.

# REFERENCIAS

[1] Unity Technologies, “Unity Multiplayer”

Fecha de consulta: 28/07/2019

<https://unity3d.com/es/unity/features/multiplayer>

[2] Varios Autores, “C Sharp”

Fecha de consulta: 28/07/2019

<https://es.wikipedia.org/wiki/C_Sharp>

[3] Unity Asset Store, “Free Platform Game Assets”

Fecha de consulta: 15/05/2019

<https://assetstore.unity.com/packages/2d/environments/free-platform-game-assets-85838>