

Estudio de un desarrollo

Grado en Ingeniería Multimedia



Trabajo Fin de Grado

Autor:

Jorge García Valera

Tutor/es:

Mireia Luisa Sempere

Mayo 2019



# MOTIVACIÓN Y JUSTIFICACIÓN

La idea de este proyecto nace con la visión de la autosuperación y el crecimiento personal dentro del ámbito del desarrollo de videojuegos.

La realización de este proyecto se plantea de tal forma en que se rehará un antiguo proyecto, el cual cuando se desarrollo no se tenia mucha experiencia en el ámbito del desarrollo de videojuegos. Ahora con más experiencia en este campo se pretende mejorar el resultado del producto final.

Para conseguir esta mejora se revisará el proyecto anterior y se extraerán las ideas principales. Una vez extraídas las ideas se procederá a su organización, es decir, decidir cual es el orden lógico en el que se deben de implementar para que el desarrollo del videojuego crezca de manera uniforme.

A su vez, para que el juego crezca de forma uniforme y con la mayor optimización posible se van a implementar varios patrones de diseño (Singleton, States, Factory…) y optimizaciones varias como un QuadTree que nos permitirá reducir considerablemente el cálculo de colisiones.

La idea para este proyecto en mi opinión es original, ya que se sale de la típica idea de únicamente desarrollar un videojuego para demostrar tus conocimientos. Esta idea puede servir para que generaciones futuras no le teman al desarrollo, ya que en sus inicios afrontar un desarrollo de un videojuego puede parecer algo imposible, pero como todo, es cuestión de dedicarle tiempo y mejorar en el entendimiento del lenguaje y de la librería que decidas utilizar. Todo esto estará recopilado al final de documento, ya que se va a estudiar el resultado de ambos proyectos.

A su vez, sirve para que el desarrollador vea como ha ido cambiado y mejorando su forma de programar y de afrontar los problemas que surgen durante un desarrollo.

# OBJETIVOS

# AGRADECIMIENTOS

# DEEDICATORIA

# CITAS

Cada época tiene su forma de contar historias, y el videojuego es una gran parte de nuestra cultura. Puedes ignorar los videojuegos o aceptarlos y empaparte de su gran calidad artística. Algunas personas están cautivadas con los videojuegos de la misma forma que a otras personas les encanta el cine o el teatro.

Andy Serkis

El objetivo claro de los videojuegos es entretener a la gente sorprendiéndoles con nuevas experiencias.

Shigeru Miyamoto

INDICE DE CONTENIDOS

# INTRODUCCIÓN

¿Qué es un videojuego Hack and Slash?

Un videojuego del género **Hack and Slash** es un videojuego en el cual se enfatiza el combate cuerpo a cuerpo. En este género, el jugador se enfrenta en un mapa lineal a una gran cantidad de enemigos. Una vez que el jugador haya derrotado a todos los enemigos y llegue al final del nivel le esperará la lucha contra el jefe final, el cual supondrá un mayor desafío.

A su vez, existe otro género que podría ser considerado el padre de este género. El género del que hablamos es **Beat ‘em up** (también conocido como **Brawler**) en el cual el jugador se enfrenta también a una gran cantidad de enemigos.

Como podemos observar, ambos géneros son muy similares, por lo cual vamos a ver cuál es la principal diferencia entre ambos. La principal diferencia entre estos ambos géneros es que en el **Beat ‘em up** el estilo de lucha es **cuerpo a cuerpo sin armas**, directamente con los puños y/o piernas. Sin embargo, en el **Hack and Slash** el protagonista suele usar un **arma blanca**, ya sea para golpear y/o lanzar.

A su vez, existe otro subgénero llamado **Shoot ‘em up** el cual está enfocado a la lucha desenfrenada empuñando **armas de fuego**.

Otra característica que comparten estos géneros es que además de ser jugados en modo de un jugador pueden ser jugados en **modo cooperativo** en dos formas distintas.

* **Pantalla compartida:** en este modo dos jugadores pueden jugar a la vez desde una misma pantalla.
* **Modo online:** en este modo cuatro jugadores pueden jugar a la vez, cada uno desde su respectiva consola.

Lo interesante de este modo cooperativo es que los jugadores no tienen que luchar o competir entre sí para ver quién es mejor, sino que deben de unir fuerzas para derrotar a las hordas y así poder llegar aún más lejos en las infinitas rondas de enemigos que plantean estos géneros.

Actualidad de los videojuegos Hack and Slash

Actualmente, el género ha sido un poco reinventado ya que hoy en día hay consolas más potentes que las que había cuando este género fue creado. A su vez, el género ha sido combinados con otros como el de Plataformas, Aventura o Acción para dotar al producto final de una mayor diversión y jugabilidad juntando lo mejor de ambos géneros.

* **For Honor:** Mezcla los géneros de Acción y Hack and Slash con una temática medieval. For Honor trata de una lucha entre 2 equipos compuestos por 3 jugadores cada uno en el cual los jugadores tienen que luchar entre sí para conquistar puestos estratégicos para que su equipo consiga puntos y gane la partida.
* **Dead Rising (franquicia):** Mezcla los géneros de Acción, Aventura, Hack and Slash y Shoot ‘em up con temática zombi. Dead Rising 4 es un juego en el cual encarnamos a Frank West y debemos de sobrevivir a los infinitos zombis recolectando y creando armas mientras avanzamos en la historia.
* **God of War (franquicia):**

# MARCO TEÓRICO / ESTADO DEL ARTE

# OBJETIVOS

# METODOLOGÍA

# ¿¿DONDE DEBERIA DE METER ESTO?? HERAMIENTAS??

Para poder realizar este proyecto se han utilizado diversos tipos herramientas, cada una especializada en un campo en concreto.

**GitKraken:** GitKraken es un gestor de versiones que nos permite gestionar nuestras versiones que tenemos en GitHub de forma gráfica, lo que permite que estas operaciones se hagan de forma más simple y rápida.

**Visual Studio Commuity 2017:** Visual Studio ha sido el IDE utilizado para desarrollar el proyecto. Se ha decidido utilizar este IDE ya que nos facilita mucho el trabajo con atajos de teclado, compilador integrado, autocompletar, texto predictivo, creación de clases con dos clics y muchas otras funcionalidades.

**SFML:** Simple and Fast Multimedia Library o SFML, es la librería que ha hecho posible el desarrollo de este proyecto junto con el lenguaje C++. Esto es gracias a que nos permite crear una ventana en la cual nosotros podemos pintar los assets (sprites, imágenes y más) de nuestro proyecto. También nos permite recoger los inputs tanto de teclado y ratón como los de un joystick para así poder interaccionar con los assets.

**Photoshop:** Photoshop es un programa de creación y edición de imágenes. Este programa nos permite crear los múltiples assets necesarios para nuestro proyecto. Más en concreto me ha permitido crear los SpriteSheets (imagen que contiene todas las texturas de una o varias animaciones) necesarios para luego convertirlos en texturas y así poder aplicarlos a los sprites dentro del videojuegos.

**Tiled:** Tiled nos permite crear de forma rápida y simple mapas en 2D con múltiples capas para luego utilizarlos en nuestros proyectos. Para poder usar Tiled primero tenemos que crear un SpriteSheet. Una vez tenemos creado el SpriteSheet lo cargamos en Tiled y con unos cuantos clics podremos crear nuestro mapa.

**TinyXML-2:** TinyXML-2 es una librería que nos permite leer el contenido del XML generado por Tiled. Cuando creamos un mapa con Tiled este se guarda en formato XML, por lo cual no podemos usarlo directamente en nuestro proyecto, es por eso por lo que necesitamos esta librería para poder sacarle partido a Tiled.

**Toggl:** Toggl es una aplicación web que nos permite llevar un registro de las horas que hemos usado para la creación del proyecto. Es tan fácil como rellenar un campo y darle al play para que empiece a contar el tiempo.

**Trello:** Trello es una aplicación web que nos permite organizar información en tarjetas las cuales podemos mover con total libertad entre las múltiples columnas que nos permite crear. Trello ha sido utilizado principalmente para crear el orden de desarrollo.

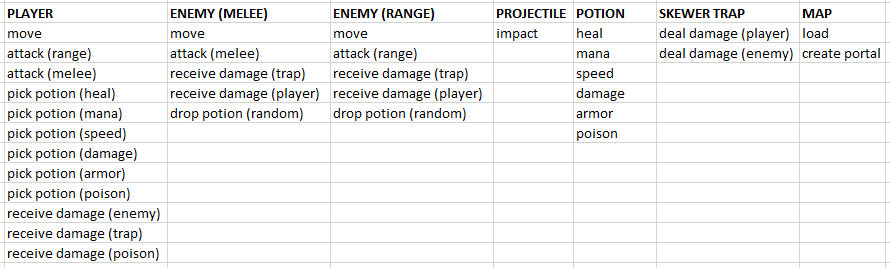
**Spotify:** Spotify es una aplicación que nos ofrece un servicio de música en streaming. Su principal uso ha sido para hacer mas amenas las sesiones de desarrollo.

# DIARIO DE DESARROLLO

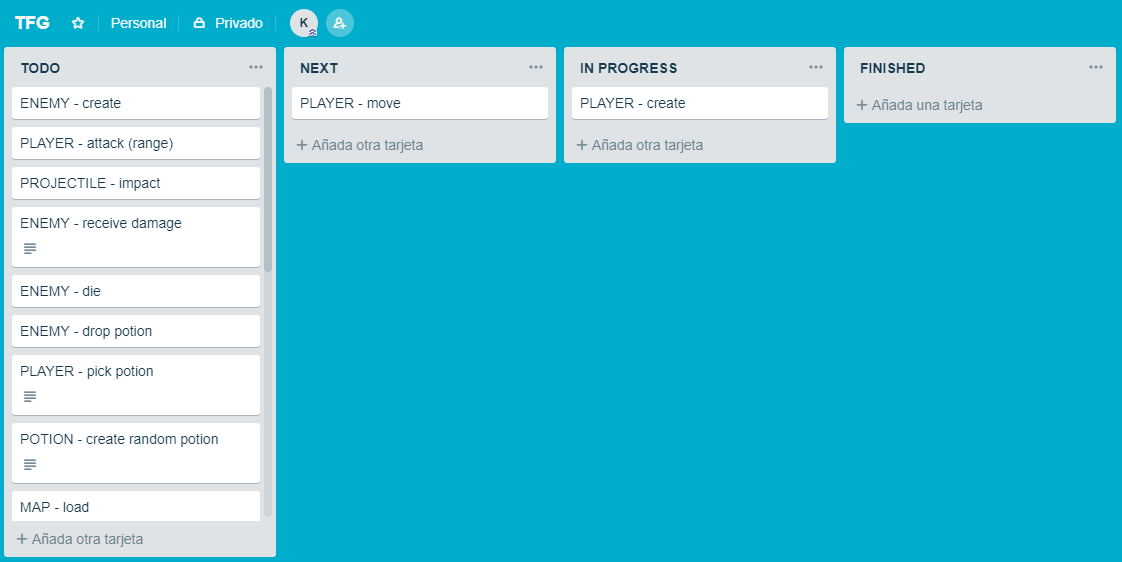
Ahora vamos a proceder a explicar como ha ido creciendo el proyecto, desde sus inicios hasta su fase final. La forma en la que se va a explicar va a se r la siguiente: la tarea se resume en un título y a su lado se pondrá el número total de horas dedicadas a ese paso y justo debajo se explicara que es lo que se ha hecho en ese paso.

**Recolectar ideas y mecánicas básicas – 2:08 horas**

Antes de empezar con el desarrollo decidí estudiar el documento de desarrollo del proyecto anterior. El estudio de este documento me permitió refrescar algunos conceptos que se me habían perdido por el camino y a su vez mejorar algunos que no terminaban de convencerme.



Una vez todo recolectado me planteé cual seria el orden lógico de implementación de dichas mecánicas. Para poder ordenar, mover y crear el orden decidí utilizar Trello ya que me permite mover las tarjetas como si de Post-It en una pizarra se tratara. Tras un largo debate mental llegue a la conclusión de cual seria la mejor manera de implementar paso a paso el proyecto, aun sabiendo que la lista en un futuro seria modificada, ya sea por cambio de orden o por adición de nuevas mecánicas o mejoras.



**Configurar Visual Studio para el desarrollo – 2:36 horas**

Una vez organizadas las mecánicas básicas empecé a configurar Visual Studio para el desarrollo. Lo primero que hice fue descargar SFML y vincular VS con SFML para poder usar la librería. Una vez lo tenia todo vinculado probé el ejemplo de SFML para ver que todo funcionaba correctamente.

**Inicio del desarrollo: crear personaje – 2:14 horas**

Una vez todo en marcha y funcionando sin problemas se empezó con el desarrollo, pero antes de ello diseñe un diagrama de clases para ver como se iban a gestionar las relaciones entre clases para facilitar el trabajo.

**INSERTAR DIAGRAMA DE CLASES**

Una vez terminado el diagrama se empezó ya con el desarrollo. Se creo un simple bucle de juego que nos permitía ver por pantalla las entidades que deseábamos. Cuando ya se mostraban por pantalla se procedió a gestionar la entrada de inputs para dotar al personaje de movimiento. Una vez implementado el movimiento se creó la cámara de seguimiento para que el personaje siempre esté centrado en la escena.

**\*\*\*\*\* AÑADIR A TOGGL 2:30 HORAS DE TRABAJOOOOR!! \*\*\*\*\***