

Estudio de un desarrollo

Grado en Ingeniería Multimedia



Trabajo Fin de Grado

Autor:

Jorge García Valera

Tutor/es:

Mireia Luisa Sempere

Mayo 2019



# AGRADECIMIENTOS

# DEDICATORIA

# 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

# INDICE

[AGRADECIMIENTOS 2](#_Toc534628577)

[DEDICATORIA 3](#_Toc534628578)

[CITAS 4](#_Toc534628579)

[INDICE 6](#_Toc534628580)

[INDICE DE FIGURAS 7](#_Toc534628581)

[INTRODUCCIÓN 8](#_Toc534628582)

[JUSTIFICACIÓN Y OBJETIVOS 9](#_Toc534628583)

[METODOLOGÍA 10](#_Toc534628584)

[MARCO TEÓRICO 12](#_Toc534628585)

[¿Qué es un videojuego Hack and Slash? 12](#_Toc534628586)

[Actualidad de los videojuegos Hack and Slash 12](#_Toc534628587)

[HERAMIENTAS 14](#_Toc534628588)

[DE GESTIÓN 14](#_Toc534628589)

[DE DESARROLLO 14](#_Toc534628590)

[GRÁFICAS 15](#_Toc534628591)

[OTRAS 15](#_Toc534628592)

[PRIMEROS PASOS 16](#_Toc534628593)

[DIARIO DE DESARROLLO 20](#_Toc534628594)

# INDICE DE FIGURAS

[Ilustración 1 - Ejemplo de Kanban 10](#_Toc534628595)

[Ilustración 2 - Ejemplo de Trello a mitad de desarrollo 11](#_Toc534628596)

[Ilustración 3 – Sprite sheet de entidades 16](file:///C:\Users\krudo\Desktop\TFG\Memoria%20TFG.docx#_Toc534628597)

[Ilustración 4 – sprite sheet de mapa 16](file:///C:\Users\krudo\Desktop\TFG\Memoria%20TFG.docx#_Toc534628598)

[Ilustración 5 - Paleta de colores 16](#_Toc534628599)

[Ilustración 6 - Menú principal 17](#_Toc534628600)

[Ilustración 7 - Menú selección modo de juego 17](#_Toc534628601)

[Ilustración 8 - Menú opciones 17](#_Toc534628602)

[Ilustración 9 - Menú selección jugador 18](#_Toc534628603)

[Ilustración 10 - Interfaz 1 jugador 18](#_Toc534628604)

[Ilustración 11 - Interfaz 2 jugadores 18](#_Toc534628605)

# INTRODUCCIÓN

La industria de los videojuegos es hoy en día una de las más grandes, superando incluso a la del cine. Es por eso por lo que cada día salen cientos de videojuegos, ya sean de grandes, medianas o pequeñas empresas, o hasta incluso de un grupo de gente que se junta para desarrollar y crear contenido en una de sus pasiones.

Ya sea por el interés económico o por crear contenido por una pasión, mucha gente se lanza a la industria del videojuego ya sea como desarrollador o como artista (2D o 3D). Una vez dentro de la industria ven que no es tan fácil llegar a lo mas alto de forma rápida, sino que tienes que empezar por el escalafón mas bajo y poco a poco ir subiendo e ir ganando experiencia.

# JUSTIFICACIÓN Y OBJETIVOS

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

La realización de este trabajo se plantea de tal forma en que se volverá a desarrollar un antiguo proyecto, el cual cuando se desarrollo no se tenia mucha experiencia en el ámbito del desarrollo de videojuegos ni con él lenguaje de desarrollo. Ahora con más experiencia en ambos campos se pretende mejorar el resultado del producto final.

Para conseguir esta mejora se revisará el proyecto antiguo y se extraerán las ideas y mecánicas principales. Una vez extraídas las ideas y mecánicas se entrará en un proceso en el cual se decidirá si esa mecánica que se pensó en el inicio es valida o no, o si necesita una mejora para que se adapte todavía mas al producto final que deseamos.

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.

Los objetivos por tratar en este trabajo son:

* Extraer las ideas y mecánicas originales y analizarlas para comprobar que son validas para el nuevo proyecto.
* Con las nuevas mecánicas e ideas desarrollar una nueva versión del videojuego original.
* Comparar ambas versiones y observar el cambio de rendimiento y de limpieza en el código.

# METODOLOGÍA

La metodología utilizada para realizar este proyecto se denomina Kanban.

La metodología Kanban consiste en un panel en el cual tenemos tantas columnas como fases queramos o necesitemos en nuestro proyecto (Por hacer, en proceso, en diseño, terminado, etc.).

Las columnas deberán ser rellenadas con tarjetas (normalmente post-it de diversos colores para una fácil y rápida interpretación del estado del proyecto), en las cuales se escribe el nombre de la tarea a realizar y una duración estimada de la misma.

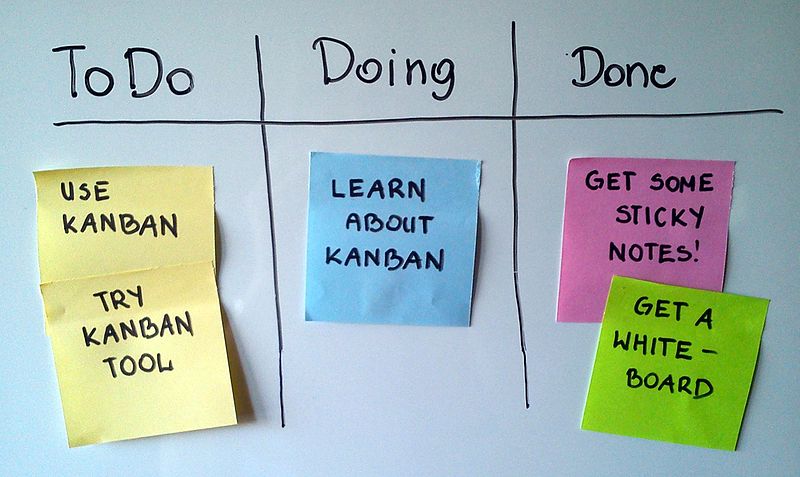


Ilustración - Ejemplo de Kanban

Antes de empezar con el desarrollo y teniendo claro que es lo que tenemos que hacer para terminar nuestro producto, tenemos que empezar por rellenar todas las tarjetas necesarias, ordenarlas por orden de prioridad (si fuera necesario) y colocarlas en la columna de ‘Por hacer’. Cuando se empiece con el desarrollo. Lo primero que tenemos que hacer si no tenemos ninguna tarea asignada es seleccionar una tarjeta y ponerla en la columna de ‘En proceso’. Una vez tengamos la tarea terminada la ponemos en la columna de ‘Terminado’ y se repetiría todo el proceso desde la primera columna.

Si durante el desarrollo nos encontramos que una tarea es demasiado grande se puede subdividir esa tarea en otras más pequeñas añadiendo nuevas tarjetas a la columna de ‘Por hacer’. De esta forma conseguiremos agilizar el flujo de trabajo y conseguir que el producto tenga una calidad superior.

Para poner en marcha esta metodología se ha decidido utilizar la aplicación Trello, que es una aplicación web que nos permite crear las columnas y las tarjetas. Además, a las tarjetas se les puede añadir funcionalidades extras como una checklist (lista con tics para marcar que se ha realizado) y añadir comentarios a las tarjetas por si nos encontramos con alguna dificultad o queremos dejarlo algo remarcado para un futuro.

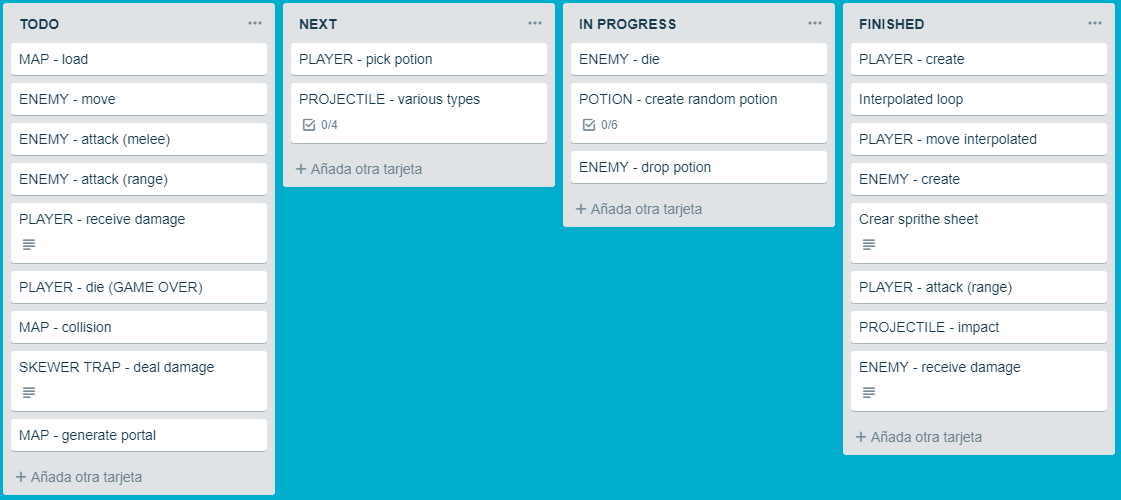


Ilustración - Ejemplo de Trello a mitad de desarrollo

# MARCO TEÓRICO

¿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.

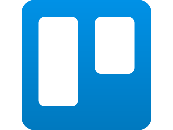
**God of War (franquicia):**

# HERAMIENTAS

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

## DE GESTIÓN

Las herramientas de gestión son aquellas que nos ayudan durante el desarrollo, ya sea mediante la organización o creando copias de seguridad por si en algún momento tenemos que retroceder debido a algún error o cambio en la estructura. Las herramientas de este tipo que hemos utilizado han sido:

**Trello:** Trello es una aplicación web que nos permite organizar información en forma de 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.

**Toggl:** Toggl es una aplicación web que nos permite llevar un registro de las horas que hemos empleado para la realización del proyecto. Para utilizarla basta con rellenar un campo con el nombre de la tarea y a continuación darle al botón de “Play” y el tiempo empezará a contarse.

**GitKraken:** GitKraken es una interfaz gráfica para git, lo que nos permite realizar en cuestión de unos pocos clics copias de seguridad de nuestro código a nuestra cuenta de GitHub.

## DE DESARROLLO

Las herramientas de desarrollo son aquellas que nos han permitido dar forma a nuestro proyecto, ya sea mediante un editor de texto o mediante librerías para añadir funcionalidad. Las herramientas de este tipo que hemos utilizado han sido:

**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.

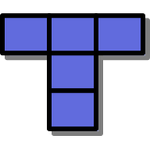
**SFML: ‘**Simple and Fast Multimedia Library’, es la librería que ha hecho posible el desarrollo de este proyecto junto con el lenguaje C++. Esto es gracias a que nos permite crear una ventana en la cual nosotros podemos pintar los assets (sprites, imágenes y más) de nuestro proyecto. También nos permite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

**TinyXML-2:** TinyXML-2 es una librería que nos permite leer de forma rápida y sencilla el contenido de un fichero XML. Esta librería la usaremos para leer el fichero XML que nos generará Tiled con toda la información del mapa del juego.

## GRÁFICAS

Las herramientas gráficas son aquellas que nos han permitido crear todos los assets visuales necesarios para poder realizar el proyecto, ya sean imágenes sueltas o ‘Sprite sheets’ (conjunto de imágenes o animaciones de un sprite).

**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.

## OTRAS

**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.

# PRIMEROS PASOS

Antes de empezar con el desarrollo y una vez establecidas las herramientas principales para poder llevar a cabo el proyecto se empezó con la búsqueda y recuperación de los assets que formaron el juego original (principalmente se recuperaron los sprites de los personajes, los enemigos y los tiles que forman el mapa).

Una vez teniendo todos los assets localizados se empezó por seleccionar cuales iban a estar presentes en esta nueva versión. A su vez se crearon algunos nuevos que no estaban en la versión original del juego y otros fueron ligeramente modificados para que encajaran mejor en esta nueva versión.

Con todos los sprites ya listos se crearon 2 sprite sheets: uno para el mapa y otro para el resto de las entidades.

Ilustración 3 – Sprite sheet de entidades

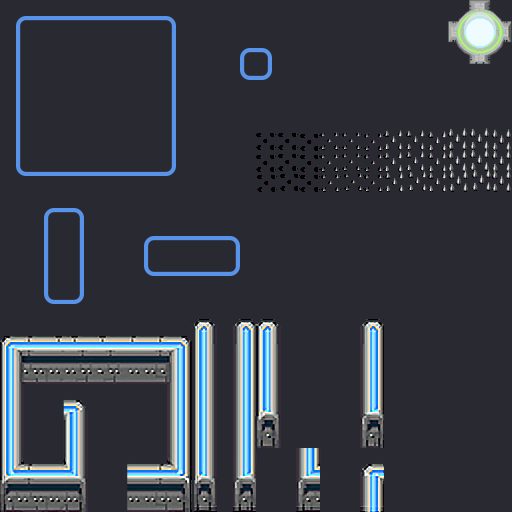


Ilustración 4 – sprite sheet de mapa

Tras observar como quedaron los sprite sheets se continuó creando una paleta de colores con los principales colores que componen la mayoría de los elementos para así poder ser usados en el menú y en la interfaz del juego.



Ilustración 5 - Paleta de colores

Cuando ya se tubo la paleta de colores completa, se procedió a crear los bocetos de los menús y el de la interfaz, para así agilizar luego el proceso de creación de los mismo y no tener que para el desarrollo para bocetarlos y posteriormente crearlos.



Ilustración 6 - Menú principal



Ilustración 7 - Menú selección modo de juego

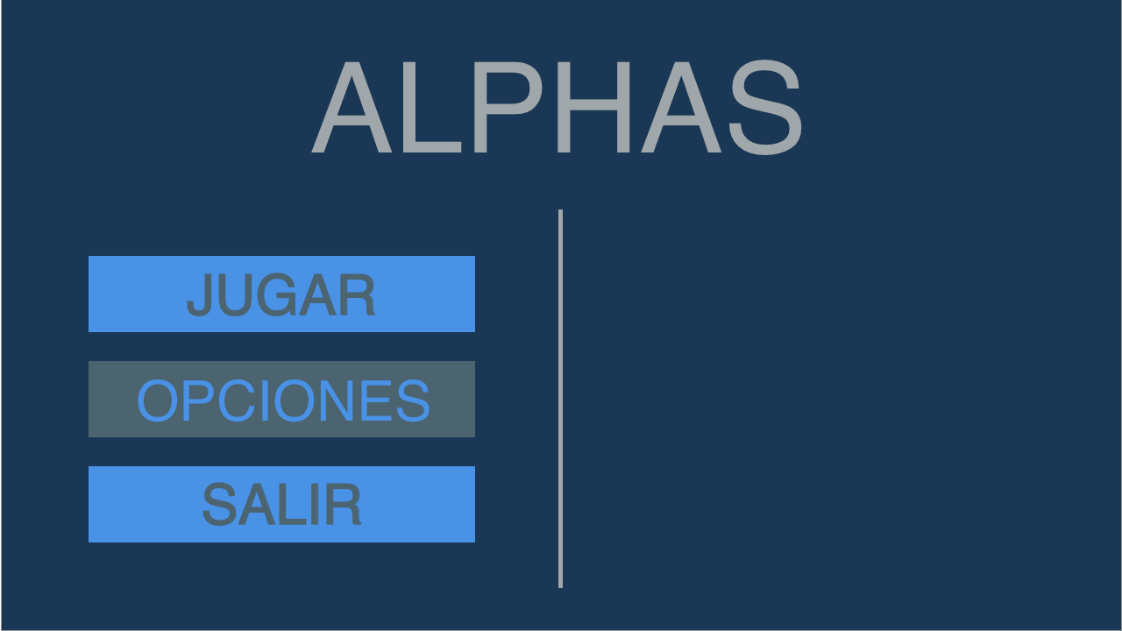


Ilustración 8 - Menú opciones



Ilustración 9 - Menú selección jugador



Ilustración 10 - Interfaz 1 jugador



Ilustración 11 - Interfaz 2 jugadores

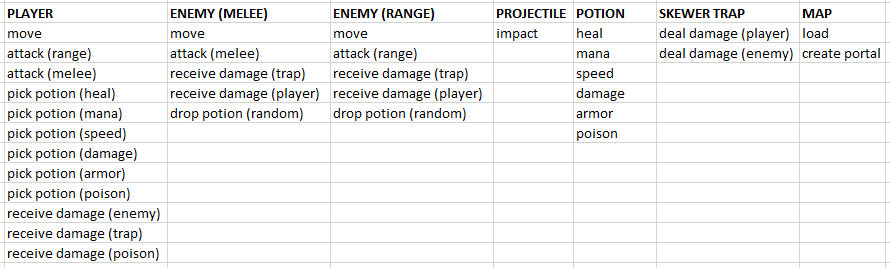
PARA EL PROXIMO DIA: CONTINUAR EXPLICANDO UN POCO MAS LOS MENUS Y LA INTERFAZ DEL JUEGO.

# DIARIO DE DESARROLLO

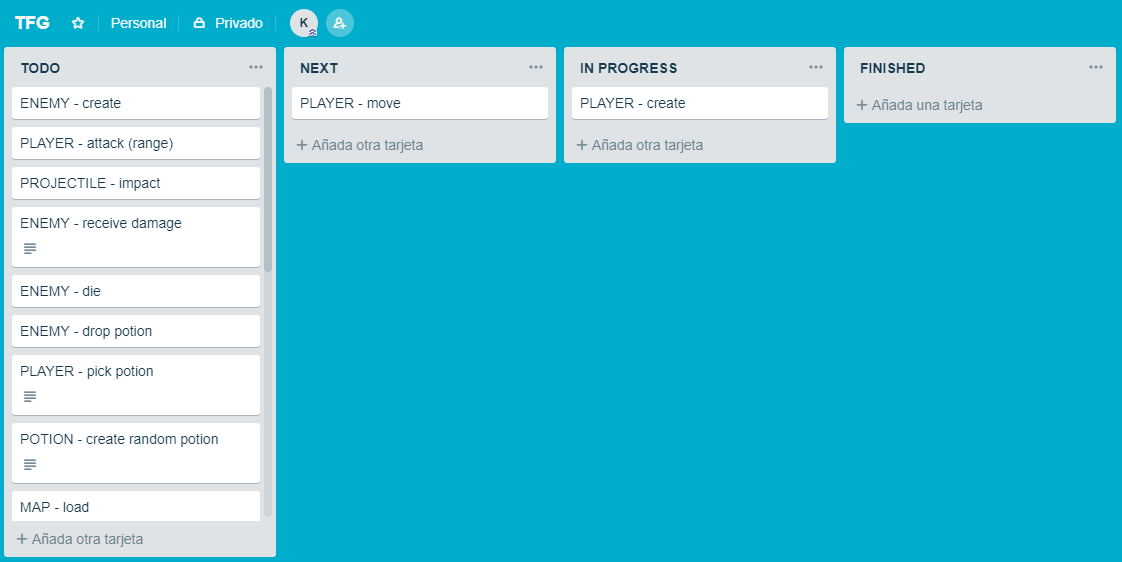
Ahora vamos a proceder a explicar como ha ido creciendo el proyecto, desde sus inicios hasta su fase final. La forma en la que se va a explicar va a se r la siguiente: la tarea se resume en un título y a su lado se pondrá el número total de horas dedicadas a ese paso y justo debajo se explicara que es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