



Universidad Politécnica
de Madrid

**Escuela Técnica Superior de
Ingenieros Informáticos**



Grado en Ingeniería Informática

Trabajo Fin de Grado

**MiJo: Videojuego para Móvil Basado en
la Estrategia**

Autor: Miguel Velasco López

Tutor(a): José Antonio Calvo-Manzano Villalón

Madrid, Junio 2019

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado

Grado en Ingeniería Informática

Título: MiJo: Videojuego Móvil Basado en la Estrategia

Junio,2020

Autor: Miguel Velasco López

Tutor:

José Antonio Calvo-Manzano Villalón

LENGUAJES Y SISTEMAS INFORMÁTICOS E INGENIERÍA DE
SOFTWARE

ETSI Informáticos

Universidad Politécnica de Madrid

Resumen

Este proyecto consiste en el aprendizaje que hay tras el desarrollo de un videojuego, donde trataremos de comprender todos los procesos involucrados en el desarrollo. También intentaremos comprender las principales diferencias en las interfaces entre la plataforma de consolas y móviles. Por último, hablaremos de las metodologías ágiles y cual hemos aplicado al trabajo.

También intentaremos familiarizarnos con las herramientas de desarrollo del entorno de los videojuegos.

El juego pertenecerá al género de los Auto Battle, un nuevo género que tuvo una explosión de popularidad en el año 2019, haremos un repaso de sus principales características y de cómo se va a adaptar a nuestro proyecto.

Abstract

This project shows the background study behind a videogame development, where we will have an insight into the aforementioned process. Furthermore, we will also take a look over the main differences in the interfaces between console and mobile platform. Finally, there is a brief discussion about the various methodologies available and the most appropriate for this project.

There is also an introduction to some of the most common development tools of the game environment.

The genre of the game is Auto Battle, a new genre that has burst since 2019, there is a review about the main features and how is depicted in the project.

Tabla de contenidos

Resumen.....	i
Abstract.....	ii
Tabla de Ilustraciones.....	v
1 Introducción	1
1.1 Contexto del sector de los videojuegos	1
1.2 Objetivos	2
1.3 Aproximación a la solución	2
1.4 Estructura del TFG.....	3
2 Estado de la cuestión.....	4
2.1 Tecnologías	4
2.2 Videojuegos de móvil populares.....	5
2.3 Género auto battler.....	6
3 Metodología	9
3.1 Metodologías tradicionales	9
3.1.1 Metodología RUP	9
3.1.2 Metodología MSF	10
3.2 Metodologías ágiles.....	11
3.2.1 SCRUM.....	12
3.3 Metodología elegida.....	13
4 Análisis de la interfaz de los videojuegos	14
4.1 Características de las interfaces de consola	14
4.1.1 Interfaz de videojuegos de disparos en primera persona	14
4.1.2 Interfaz de videojuegos de conducción	15
4.2 Características de las interfaces de móvil.....	16
4.2.1 Interfaz de videojuegos de disparos en primera persona	16
4.2.2 Interfaz de videojuegos de conducción	17
4.3 Diferencias	17
5 Desarrollo	19
5.1 Entorno de desarrollo	19
5.1.1 Unity	19
5.1.2 Visual studio.....	23
5.1.3 Blender.....	23
5.2 Gameplay.....	28
5.3 Proceso de desarrollo del juego.....	29
6 Conclusiones y líneas futuras.....	33
6.1 Principales dificultades encontradas.....	33
6.2 Mejoras y líneas futuras.....	33

6.3	Conclusiones.....	33
7	Bibliografía.....	35

Tabla de Ilustraciones

Ilustración 1: La consola NES.....	1
Ilustración 2: Playstation one	2
Ilustración 3: Logo de Unreal Engine	4
Ilustración 4: Logo de Unity.....	4
Ilustración 5: Logo de candy Crush	5
Ilustración 6: Imagen del juego Pokemon go.....	5
Ilustración 7: Imagen del juego Clash of Clans	6
Ilustración 8: Imagen del modo de juego de DOTA 2 Auto Chess	7
Ilustración 9: Imagen del juego Auto Chess desarrollado por Drod Studio.....	7
Ilustración 10: Fases RUP	10
Ilustración 11: Fases MSF.....	11
Ilustración 12: HUD FPS	15
Ilustración 13: HUD de juegos de conducción	15
Ilustración 14: HUD FPS móvil.....	16
Ilustración 15: HUD juego de conducción para móvil.....	17
Ilustración 16: vista de la jerarquía	20
Ilustración 17: vista de la escena.....	20
Ilustración 18: Carpeta de proyecto	21
Ilustración 19: Vista del juego	21
Ilustración 20: Inspector	22
Ilustración 21: Vista de la consola	22
Ilustración 22: Visual Studio	23
Ilustración 23: C#	23
Ilustración 24: Layout.....	24
Ilustración 25: Modelado	24
Ilustración 26: Esculpir.....	25
Ilustración 27: Edición UV	25
Ilustración 28: Pintar las texturas	26
Ilustración 29: Sombreado	26
Ilustración 30: Animación	27
Ilustración 31: Renderizado.....	27
Ilustración 32: Pantalla de scripting	28
Ilustración 33: Imagen de campo de batalla	29
Ilustración 34: Concepto de unidad	30
Ilustración 35: Algunos componentes de la unidad.....	31
Ilustración 36: Panel de la tienda con alguno de sus componentes.....	32

1 Introducción

El sector de los videojuegos hoy en día es una de los más influyentes en nuestra sociedad, no solo a nivel de entretenimiento sino también desde el punto de vista artístico y económico, y no es de extrañar puesto que permite un abanico de posibilidades y mundos infinitos para que las personas puedan decidir como entretenerse.

El sector de los videojuegos sigue al alza en crecimiento y no parece que vaya a detenerse pronto. Dentro del sector de los videojuegos, el mercado de móviles ha sido el que ha experimentado un crecimiento más rápido en los últimos años, hasta convertirse en el más grande del sector, todo ello debido a la llegada de los smartphones y la capacidad de llevarte los videojuegos a cualquier parte y jugarlos en cualquier momento.

1.1 Contexto del sector de los videojuegos

Los primeros videojuegos surgieron en la década de los cincuenta, poco tenían que ver aquellos videojuegos con los de la actualidad, los recursos que había eran claramente distintos, acababan de aparecer los primeros ordenadores, y mientras ahora los videojuegos populares son conocidos por prácticamente todo el mundo, en aquella época no eran casi conocidos y solo se movían en ciertos círculos, como el universitario.

En la década de los setenta surgió el primer sistema con varios juegos pregrabados, la Magnavox Odyssey desarrollada por Ralph Baer, esta consola que permitiría jugar a varios videojuegos en un mismo dispositivo y comercializarlos de una manera más sencilla. En esta década también se popularizaron las máquinas recreativas que se introdujeron en distintos lugares públicos como bares, salones recreativos o aeropuertos. Durante esta década se lanzarían múltiples consolas y videojuegos clásicos como el pacman.

En los años ochenta Estados Unidos y Canadá sufrieron una crisis respecto a los videojuegos mientras que Japón y Europa se dedicaron a desarrollar sus propias consolas como las NES por parte de Japón y el Spectrum en Europa el cual era un microordenador.



Ilustración 1: La consola NES

A principio de los noventa las consolas, conocidas como generación de los 16 bits y estaba formada por consolas como la SNES o la Mega Drive, dieron un gran salto técnico respecto a lo que ya había. Estas consolas consiguieron un gran aumento en cuenta al número de usuarios de consolas y también ayudaron a la introducción de

nuevas tecnologías como el CD-ROM. También se empezaron a diseñar los primeros juegos tridimensionales que más tarde acapararían el mercado.

Durante esta década también salieron las primeras consolas portátiles como la Game Boy que fue todo un éxito. A finales de la década la consola más popular era la Playstation con juegos mundialmente conocidos como el Final Fantasy, saga la cual perdurará hasta nuestros días.



Ilustración 2: Playstation one

A partir de aquí los cambios serían mucho más acelerados y se introducirían nuevas tecnologías a una mayor velocidad, también se empezarían a crear juegos multijugadores gracias a internet que permitía una fácil conexión entre ordenadores.

El siguiente gran cambio se produciría con la séptima generación de consolas en 2006, esta generación aparte de su gran mejora en el apartado técnico respecto a los videojuegos también soportaba contenidos multimedia como DVD. [1]

Y como último paso para asentar los cimientos del mercado de videojuegos surgió el primer iPhone en 2007 lo cual supuso el salto de los videojuegos a los smartphones.

Desde 2007 hasta la actualidad han surgido una gran cantidad de mejoras e innovaciones tecnológicas, pero también ha crecido el mercado de los videojuegos llegando a mover cifras económicas de más de 91.000 millones de dólares de los cuales 40.600 millones de dólares pertenecen al sector de los móviles. Y se espera que el mercado de los videojuegos siga creciendo al menos un 10% por año.

1.2 Objetivos

El proyecto consiste en desarrollar un videojuego para móvil de estrategia del género autobattle, mediante este proyecto se intentará comprender el proceso de desarrollo de videojuegos de principio a fin, así como las principales diferencias de las interfaces de usuario que existen entre consolas y móviles. También se pretende aprender un poco del mercado de videojuegos para móvil en cuanto a su relevancia en el sector de los videojuegos.

1.3 Aproximación a la solución

Se pretende desarrollar un videojuego para móvil desde el inicio utilizando el motor gráfico Unity 5 con el cual tendremos que aprender a modelar 3D y a programar IA para algunos de los personajes, también durante el desarrollo de la interfaz iremos

haciendo comparaciones entre las interfaces de consola y móvil. Todo el proyecto seguirá metodologías ágiles.

1.4 Estructura del TFG

1. Introducción: Se contará un poco la historia de los videojuegos y se planteará la intención de este TFG.
2. Estado de la cuestión: En este capítulo se hablará de las tecnologías más populares y la que utilizaremos en este proyecto, también se comentaran algunos de los juegos más populares junto con las características que comparten
3. Metodología: Hablaremos sobre las distintas metodologías ágiles y como las aplicaremos a este proyecto.
4. Análisis de las interfaces de los videojuegos: Repasaremos las características más comunes de algunos géneros en consola y las compararemos con sus juegos homólogos en móvil.
5. Desarrollo: Contaremos todo el proceso de desarrollo del juego.
6. Conclusiones y líneas futuras: Realizaremos un resumen con las conclusiones y conocimientos obtenidos y realizaremos un listado de posibles mejoras que añadir al juego.

2 Estado de la cuestión

En este capítulo haremos un breve análisis de las tecnologías más populares y sus principales diferencias, echaremos un vistazo también a los juegos de móvil más populares y algunas de sus características y para finalizar explicaremos en que consiste el género de videojuegos llamado autobattle, cuáles son sus principales características, de donde surgió y como se popularizó. Apartado 1 de capítulo 2

2.1 Tecnologías

Los motores de gráficos más populares para el desarrollo de videojuegos son sin lugar a duda Unity 5 y Unreal Engine 4, son dos motores gráficos completamente gratuitos que permiten desarrollar videojuegos con resultado profesional sin ninguna dificultad.

Unreal Engine es un motor gráfico creado por la compañía Epic Games, en el apartado gráfico más potente que el Unity sin dudas, es un software pensado para usuarios experimentados que contiene herramientas avanzadas para desarrollar el videojuego y utiliza el lenguaje de programación C++.



Ilustración 3: Logo de Unreal Engine

Mientras que Unity 5 es el motor gráfico de la empresa Unity Technologies, es un software que dispone de una curva de aprendizaje sencilla, aunque gráficamente no es tan potente como Unreal Engine 4 también tiene una gran potencia gráfica, los lenguajes que utiliza con C# y JavaScript y además dispone de integración multiplataforma permitiendo trasladar de forma sencilla los juegos a Android e IOS.



Ilustración 4: Logo de Unity

El motor gráfico que se va a utilizar en este proyecto es el Unity 5 debido a una sencilla curva de aprendizaje y a su integración multiplataforma la cual nos facilitará el desarrollo de nuestro videojuego para móvil.

2.2 Videojuegos de móvil populares

En este apartado vamos a hablar de 3 de los juegos más populares para móviles: Candy Crush, Pokemon Go y Clash of Clans.

Candy Crush desarrollado por King es un videojuego de resolver puzzles en los que tendrás toda la pantalla llena de gominolas distintas las cuales podrás intercambiar entre ellas y tendrás que unir tres o más de ellas iguales para ir aumentando tu puntuación. La principal característica de este juego es su simplicidad y su rapidez permitiendo que cualquier persona en cualquier momento se ponga a intentar resolver un puzzle. Este juego cuenta con más de 500 millones de descargas y a pesar de ser un juego gratuito en 2014 llegó a ingresar alrededor de 1330 millones de dólares. [2]



Ilustración 5: Logo de candy Crush

Pokemon Go desarrollado por Niantic en colaboración con Pokémon Company crearon un videojuego de realidad aumentada en el que a través de tu smartphone podrás capturar Pokémon que se encuentran en el mundo real. Este juego se popularizó por ya que permitía de la posibilidad de encontrarte a tus Pokémon favoritos en el mundo real en cualquier parte con tan solo iniciar la aplicación y dar un pequeño paseo. Cuenta con más de 100 millones de descargas y generando más de 900 millones de dólares en 2019 siendo de uso gratuito. [3]



Ilustración 6: Imagen del juego Pokemon go

Y por último Clash of Clans desarrollado por Supercell, es un juego de estrategia en el que tienes que hacer crecer tu aldea lo máximo posible mientras tus soldados se enfrentan a las aldeas de otros jugadores. Cuenta con más de 500 millones de descargas y generó en 2015 alrededor de 1345 millones de dólares. [4]



Ilustración 7: Imagen del juego Clash of Clans

Todos estos exitosos juegos comparten ciertas características muy importantes para el sector:

- Son juegos que se pueden jugar en cualquier situación no se requiere ninguna condición importante, como una sala preparada para jugar o un espacio que permita una concentración máxima sin distracciones.
- Gráficamente no son la gran cosa, aunque sus gráficos no son malos no son los más realistas del sector.
- Sesiones cortas de juego te permiten avanzar a una buena velocidad y no necesitas estar enganchado durante horas y horas seguidas para conseguir un pequeño progreso.
- Son juegos que no requieren una gran habilidad para ser jugados, no necesitan una gran coordinación mano-ojo para poder conseguir tu objetivo.

2.3 Género Auto Battle

Este nuevo género surgió en 2019, el primer juego que obtuvo una gran fama surgió a través de una modificación del famoso juego para ordenador DOTA 2, esta modificación se llamaría Auto Chess y la creo el estudio Drodo Studio que más adelante pasaron a llamarse Dragonest.

DOTA 2 es un juego del género MOBA un juego multijugador en el que se enfrentaran dos equipos y ganará el que destruya antes la base del rival.

Una modificación de un juego es cuando el creador del contenido te permite acceder a los recursos del juego original para modificarlos y así crear nuevos personajes, modos de juego o juegos directamente. La modificación que hizo Drodo studio empezó como un modo de juego completamente distinto al DOTA 2, lo único que tendrían en común serían los personajes.

Este modo de juego llamada Auto Chess será un juego por rondas y será un todos contra todos, pero mientras que en un moba tu controlas al personaje, en este modo tus personajes se moverán solos y combatirán contra los personajes de los otros jugadores en cada ronda. En cada ronda recibirás oro, la moneda del juego, y la

utilizaras para comprar unidades o para subir de nivel, lo cual te permitirá tener más unidades en juego. Las unidades que compres tendrán distintos costes de oro, distintas estadísticas y distintas sinergias con las otras unidades, que te otorgarán ventajas o desventajas.

El jugador tendrá un contador de vida e ira perdiendo vida a medida que va perdiendo rondas. El ultimo jugador vivo gana.

Este modo de juego para mayo de 2019 consiguió tener un total de 8 millones de jugadores y unos 300.000 concurrentes. [5]



Ilustración 8: Imagen del modo de juego de DOTA 2 Auto Chess

Mas adelante Valve, creadora de DOTA 2 y Drodio Studio se separarían creando cada uno su propio Auto Battle, los dos estudios utilizarían como base el Auto Chess pero lo acabarían adaptando a sus gustos, el juego que saco Drodio Studio fue para móvil y se llamaría Auto Chess, más adelante también saldría para ordenador, mientras que el juego de Valve se llamaría Dota Underlords y saldría primero para ordenador y más adelante para móvil.



Ilustración 9: Imagen del juego Auto Chess desarrollado por Drodio Studio

Después de que el género se volviera tan popular en tan poco tiempo otras desarrolladoras se interesaron en sacar sus propios juegos de Auto Battle.

Algunas de las desarrolladoras más famosas que se animaron a sacar sus propios Auto Battles son Riot Games con su modo de juego Team Fight Tactics, conocido por sus siglas TFT, primero salió para ordenador y varios meses después salió para móviles. Actualmente el juego cuenta con 33 millones de jugadores mensuales. [6]

Otra desarrolladora famosa en lanzar su modo fue Blizzard en su juego de cartas heathstone, el modo se llama Battlegrounds.

3 Metodología

En este capítulo hablaremos de las metodologías más populares de gestión de proyectos de desarrollo de software, explicaremos los dos principales grupos y en que se diferencian y por último analizaremos la metodología seleccionada para este trabajo.

3.1 Metodologías tradicionales

Al inicio del desarrollo software no se utilizaban ningún tipo de metodologías para los procesos que hay que llevar a cabo, debido a la necesidad de mejorar los procesos y asegurarse que el desarrollo de software llegaba a la meta deseada, empezaron a implementarse los fundamentos de las metodologías de otros campos adaptándolas al desarrollo de software.

Estas metodologías se basan en etapas secuenciales muy estrictas en las que realizar alguna modificación una vez el proyecto ha empezado supondría un gran coste, también es muy importante realizar una documentación exhaustiva del proyecto. Las principales etapas de estas metodologías son:

- **Concepción:** Planteamiento inicial del proyecto, en que va a consistir y que recursos se van a utilizar.
- **Planificación:** En esta etapa se realizará dividirá el proyecto en actividades a realizar durante el desarrollo del proyecto y los recursos asociados a cada actividad.
- **Ejecución:** Se realizarán las actividades fijadas durante la planificación y se establecerán los grupos de trabajo.
- **Seguimiento y control:** El objetivo de esta fase es hacer un seguimiento de las actividades y gestionar las incidencias que se vayan produciendo.
- **Cierre del proyecto:** En esta última fase se evaluará y verificara que el proyecto cumple con las expectativas fijadas y no contiene errores, antes de darlo por concluido. [7]

3.1.1 Metodología RUP

Esta metodología es un proceso formal que pretende conseguir el correcto desarrollo de un producto software mediante un acercamiento disciplinado a la hora de asignar tareas y responsabilidades dentro del grupo que se encarga del proyecto. Fue desarrollada por Rational Software.

El objetivo de esta metodología es asegurar que la producción de software es de alta calidad y cumpa con las expectativas de los usuarios. [8]

Consta de cuatro fases:

- **Concepción**
- **Elaboración**
- **Construcción**
- **Transición**

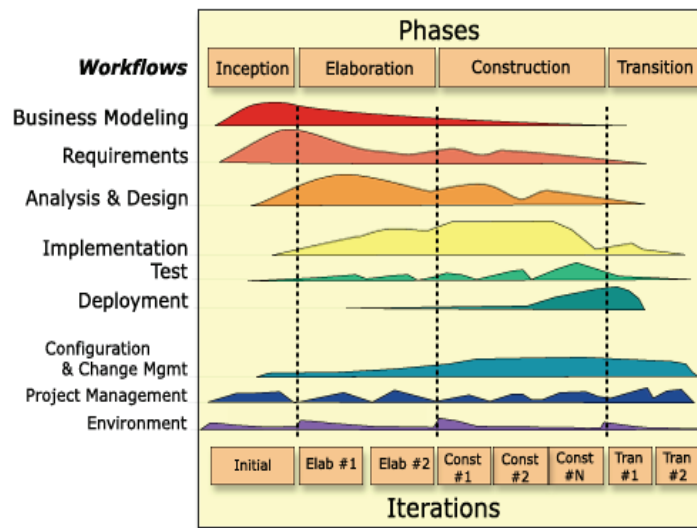


Ilustración 10: Fases RUP

Las principales ventajas son:

- Evaluación en cada fase permitiendo realizar cambios de objetivos.
- Es sencillo, sigue pasos intuitivos a la hora de desarrollar software.
- Seguimiento detallado de cada una de las fases.

Las principales desventajas son:

- La evaluación de riesgos es compleja.
- Excesiva flexibilidad para algunos proyectos.
- El cliente deberá tener un gran conocimiento de desarrollo de software para poder fijar el alcance del proyecto con él.

3.1.2 Metodología MSF

MSF no es una metodología rígida, es una recopilación de modelos que puedan adaptarse a cualquier proyecto de tecnología de la información. [8]

Las principales fases son:

- **Visión y alcance:** Es una de las fases más importantes en las que se definirán los roles del equipo y se fijara el objetivo del proyecto que tiene que haber sido acordado de manera correcta entre el cliente y el equipo.
- **Planificación:** En esta parte se planifica el proyecto, se preparan las especificaciones funcionales, el diseño, estimación de costes y los plazos de entrega.
- **Desarrollo:** En esta etapa el equipo de trabajo realizara las tareas de documentación y el código del proyecto.
- **Estabilización:** Se realizarán pruebas realistas para encontrar errores y solucionarlos.
- **Implantación:** Aquí se implanta la tecnología base y los componentes base y se estabiliza la instalación.



Ilustración 11: Fases MSF

3.2 Metodologías ágiles

Este tipo de metodología surge para solucionar los principales problemas de las metodologías tradicionales, esta metodología se basa en dos aspectos principales, retrasar las decisiones y planificación adaptativa.

El cliente deberá determinar de forma clara el objetivo y funcionamiento del producto, por lo que será necesario una comunicación fluida con el equipo de trabajo antes y durante el desarrollo del proyecto.

Se creo un manifiesto para las metodologías ágiles que consta de 12 principios que son: [9]

1. "Nuestra más alta prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor.
2. Recibe el cambio en requisitos, incluso avanzado en el desarrollo. Los procesos ágiles aprovechan el cambio para la ventaja competitiva del cliente.
3. Entregar software funcional y con frecuencia, de un par de semanas a un par de meses, con una preferencia a la escala de tiempo más corta.
4. Los empresarios y desarrolladores deben trabajar juntos a lo largo de todo el proyecto.
5. Desarrolla proyectos alrededor de personas motivadas. Bríndales el entorno y apoyo que necesiten, y confía en que hagan el trabajo.
6. El método más eficiente y efectivo para transmitir información hacia y dentro de un equipo de desarrollo es la conversación cara a cara.
7. Software funcional es la principal medida de progreso.
8. Los procesos ágiles promueven el desarrollo sostenible. Los patrocinadores, desarrolladores y usuarios deberían poder mantener un ritmo constante de manera indefinida.
9. La atención continua a la excelencia técnica y al buen diseño aumenta la agilidad.
10. La simplicidad - el arte de maximizar la cantidad de trabajo no realizado - es esencial.
11. Las mejores arquitecturas, requisitos y diseños surgen de equipos autoorganizados.
12. A intervalos regulares, que el equipo reflexione sobre cómo ser más efectivo, para sintonizar y ajustar su comportamiento".

3.2.1 SCRUM

Scrum es un proceso de gestión que reduce la complejidad en el desarrollo de productos para cumplir con las necesidades de los clientes. El desarrollo se lleva de forma iterativa, cada iteración es un ciclo en el que se termina una parte ejecutable de software que añade una nueva funcionalidad.

Este proceso se centra en priorizar el trabajo en función del valor que tengan, maximizando la utilidad de lo que se construye. Esta diseñado especialmente para

adaptarse los cambios en los requerimientos, los requerimientos y las prioridades se revisan y ajustan durante el proyecto en intervalos cortos y regulares.

3.3 Metodología elegida

La metodología elegida para este trabajo de fin de grado es la Kanban. Kanban es una palabra japonesa que está compuesta por dos partes, Kan que significa visual y Ban que significa tarjeta.

Una de las características más importantes de esta metodología es que las cosas que se tienen que hacer bien a la primera, se tarda más en arreglar los errores que asegurarse que haces las cosas bien a la primera, por ello es más importante la calidad que la rapidez.

Otra característica importante es la reducción de desperdicios, solo hay que realizar lo justo para que funcione sin distraerse en funciones extras que añadir.

La flexibilidad es muy importante en esta metodología, poder reaccionar a los imprevistos que surjan es muy relevante, de forma que siempre exista solo una cola de tareas en espera en las que hay que ir priorizando en función de las urgencias de cada una de ellas. [10]

Hemos seleccionado esta metodología porque creemos que es la que mejor se adecua al desarrollo de un videojuego realizado por una persona.

El trabajo al ser realizado por una sola persona nunca se van a poder concluir tareas de formas simultaneas, aquí la cola única de tareas que plantea Kanban resulta muy útil.

En un videojuego es muy sencillo darse cuentas de nuevas funcionalidades no necesarias que añadir por lo que ceñirse a lo estrictamente necesaria no ahorra una gran cantidad de sobre esfuerzo.

Por último, asegurarnos que las tareas realizadas hasta el momento nos ayudasen a la hora de probar nuestro juego y encontrar los errores.

La herramienta elegida para seguir esta metodología se llama Kanbantool que es de uso gratuito que nos permitirá seguir el plan de una forma más sencilla.



Ilustración 12: Aplicación Kanbantool con algunas tareas

4 Análisis de la interfaz de los videojuegos

En este capítulo vamos a analizar las características más representativas de las interfaces de consola y las interfaces de móvil. También vamos a ver sus principales diferencias y explicar las motivaciones que hay detrás de estas.

Nos centraremos mayormente en el HUD o barra de estado, que es la manera que tiene el juego de mostrarnos la información de lo que está pasando en el juego a través de la interfaz. Donde mejor podremos apreciar las diferencias de HUD entre plataformas será cuando comparemos juegos del mismo género. Para esta comparación utilizaremos dos géneros de ejemplo que serán videojuegos de disparos en primera persona, también conocidos como FPS, y juegos de conducción.

4.1 Características de las interfaces de consola

En este apartado vamos a hablar de las características más importantes de la interfaz para consola, lo haremos en dos partes una por cada genero de los videojuegos que vamos a analizar.

4.1.1 Interfaz de videojuegos de disparos en primera persona

El primer género que vamos a comentar es el de disparos en primera persona conocimos como FPS y no referiremos a ellos como FPS a partir de ahora.

Todos estos juegos tienen un indicador de salud en la pantalla, en el counter strike global offensive por ejemplo abajo a la izquierda hay una barra de salud y te muestra el porcentaje de salud que te falta, en otros juegos como el call of duty no te muestra la barra de vida, pero cuando te hacen daño te saltan indicadores rojos de que recibes daño o cuando te falta poca vida la pantalla obtiene un tono rojizo para avisarte.

Todos los FPS tienen alguna manera de mostrar donde estas para guiarte, la mayoría opta por colocar en una esquina el mini mapa en un lugar que no moleste demasiado.

Otro elemento que tienen todos en común es una zona para informarte de tu equipo, que arma llevas, cuantas balas tienes en el cargador, cuantas balas tienes para luego recargar tu arma o si tienes equipo extra. Esta información suele estar en la esquina inferior derecha de la manera que ocupe la menor cantidad de espacio y sea lo más clara posible.

Uno de los elementos más importante de esta HUD es el marcador, la mayoría de los FPS suelen ser por equipos que se enfrentan entre sí, por ello es necesario un marcador que muestre el estado de la partida, la posición de este marcador varía según el juego, pero en todos está en algún borde donde pueda molestar lo menos posible.

Y por último vamos a hablar del historial de muertes de la partida o feed, aquí se va mostrando quien mata a quien y con qué arma. Suele estar al lado del marcador y solo se muestra cuando ocurre una baja sino se esconde.



Ilustración 13: HUD FPS

4.1.2 Interfaz de videojuegos de conducción

En este género de juego la información más importante es el indicador de la velocidad viene indicado por un número y las unidades son kilómetros/hora y está acompañado por una barra que indica las revoluciones a las que va el motor. Se suele encontrar en el borde inferior de la pantalla.

Otras características importantes son las vueltas que llevas en la carrera y las vueltas suele estar en una esquina inferior.

La clasificación es otro elemento importante que muestra la información de la clasificación actual de la partida y también suele mostrar a cuantos segundos estas de tus competidores.

Por último, algunos juegos muestran el estado del vehículo, suele ser un dibujo esquematizado del vehículo con colores para informarte del estado de las partes del vehículo.



Ilustración 14: HUD de juegos de conducción

4.2 Características de las interfaces de móvil

En este apartado repasaremos las características más relevantes y comunes entre los dos géneros seleccionados en las plataformas móviles.

4.2.1 Interfaz de videojuegos de disparos en primera persona

El primer componente de HUD que nos encontramos son los botones de acciones suelen encontrarse en las esquinas inferiores a la altura de los pulgares. Estos botones sirven para realizar las acciones básicas en el lado derecho suelen estar los botones de disparar, agacharse, saltar, recargar y apuntar. En el lado izquierdo esta un botón que hace la función de joystick para poder moverse.

También tienen mini mapa para orientarse en las esquinas superiores.

El armamento que tienes se muestra lo más pequeño posible en el borde inferior.

El indicador de salud es todavía más pequeño que en consola, y para informarte que tienes poca vida funciona de la misma manera, parpadeando toda la pantalla en colores rojizos.

El marcador de la partida suele estar en una de las esquinas superiores de la pantalla.

Y por último también tiene una serie de botones para configurar ciertos aspectos del juego, como si quieres correr todo el rato, si quieres ponerte el micrófono activo o si quieres abrir el menú de opciones.



Ilustración 15: HUD FPS móvil

4.2.2 Interfaz de videojuegos de conducción

En los juegos de conducción para móviles colocan el medidor de la velocidad en el lateral derecho algunos en la esquina superior y otros en la esquina inferior. En móvil no todos llevan un indicador de las revoluciones del coche.

Otra información muy relevante que aporta el HUD es el número de vueltas de la carrera y porque vuelta vas, también tienes el tiempo que llevas en la vuelta actual.

También tienen los iconos de los botones para que sepas donde colocar los dedos para jugar, pero solo muestran al principio de la carrera durante la carrera suelen desaparecer.

Los botones que están en el lado derecho suelen ser para acelerar, frenar y realizar alguna actividad especial como activar el turbo.

Y los botones del lado izquierdo suelen ser para controlar la dirección en la que se mueve el coche.



Ilustración 16: HUD juego de conducción para móvil

4.3 Diferencias

En los géneros de disparos en primera persona la principal diferencia que nos encontramos es la ausencia de los botones con los que vas a jugar en el HUD en los juegos de consola, mientras que en los juegos de móvil es indispensable.

El historial de bajas en los juegos de móvil directamente no aparece.

En los juegos de conducción nos encontramos la misma diferencia que en los FPS en móvil se muestran los botones y en consola no. Otra gran diferencia que podemos notar es la austeridad de la información que te dan, mientras en consola te dan todo tipo de detalles de cómo va la carrera y cuál es el estado del coche, en móvil te dan la información justa para que puedas saber cómo vas en la carrera.

Las conclusiones que podemos sacar al comparar estos dos géneros de videojuegos son:

- En móvil los botones en la HUD son necesarios porque no poses un mando con el que jugar mientras que en consola no tienes este problema.
- Las HUD de consolas suelen tener mayor cantidad de detalles a la hora mostrarte la información, esto es posible porque el espacio que ocupa la información de los botones en el móvil en consola no está dejando más espacio libre.

En conclusión, podemos decir que las HUD en móvil tienen que ser más compactas y menos detallistas no solo porque la pantalla es más pequeña sino porque también los botones ocupan un espacio extra que en consola tienen libre.

5 Desarrollo

En este capítulo abarcaremos todo el desarrollo del proyecto, justificaremos la elección de las tecnologías seleccionadas y contaremos sus principales ventajas frente a sus competidores.

5.1 Entorno de desarrollo

Aquí vamos a hablar del entorno de desarrollo utilizados, como funcionan y sus características.

5.1.1 Unity

Unity es un motor de videojuegos multiplataformas, es decir, es el conjunto de herramientas que están enfocadas a desarrollar la lógica de un videojuego. Permite una edición e iteración rápida en el ciclo de desarrollo. También tiene otras utilidades fuera del mundo de los videojuegos como realizar simulaciones aprovechando su motor de físicas o para realizar interfaces gracias a sus sencillos controles.

Uno de los puntos fuertes de Unity es su comunidad, al ser muy amplia permite encontrar toda la documentación que necesites y también tiene multitud de foros en los que preguntar las dudas que tengas o encontrar preguntas más específicas.

En Unity todo se maneja desde el editor visual que es la interfaz principal y está dividida en distintas partes:

- La **vista de la jerarquía** es donde se muestran todos los elementos que tenemos en nuestra escena, como la cámara, el terreno, las luces o el jugador. En este panel podemos añadir nuevos elementos, eliminar los elementos existentes o seleccionarlos para editarlos luego, también podemos organizar la jerarquía que queremos que tengan estos objetos.

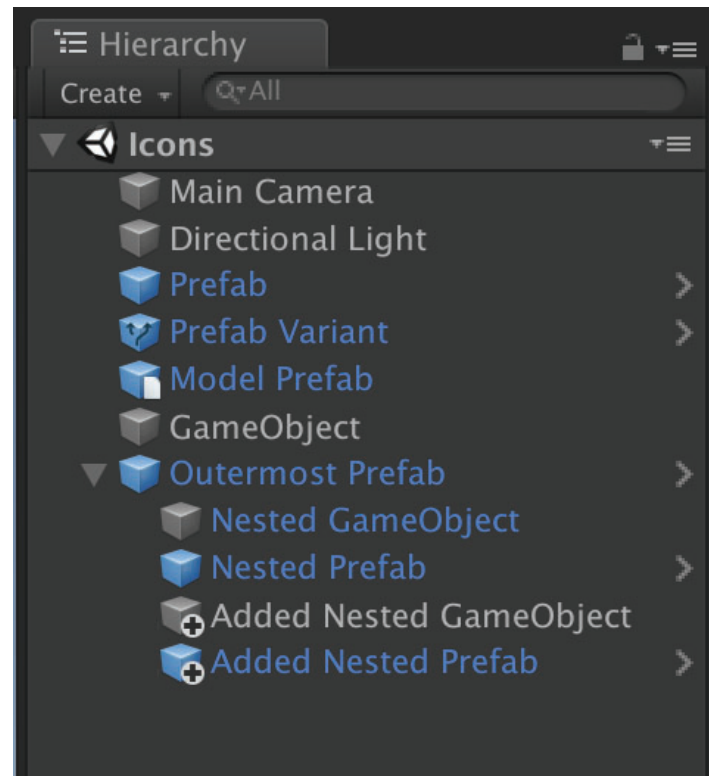


Ilustración 17: vista de la jerarquía

- La **vista de la escena** es donde puedes interactuar con mundo que estas creando, donde puedes seleccionar y colocar todos los elementos del mundo como el escenario, la cámara o las luces, también puedes cambiarles el tamaño o rotarlos.

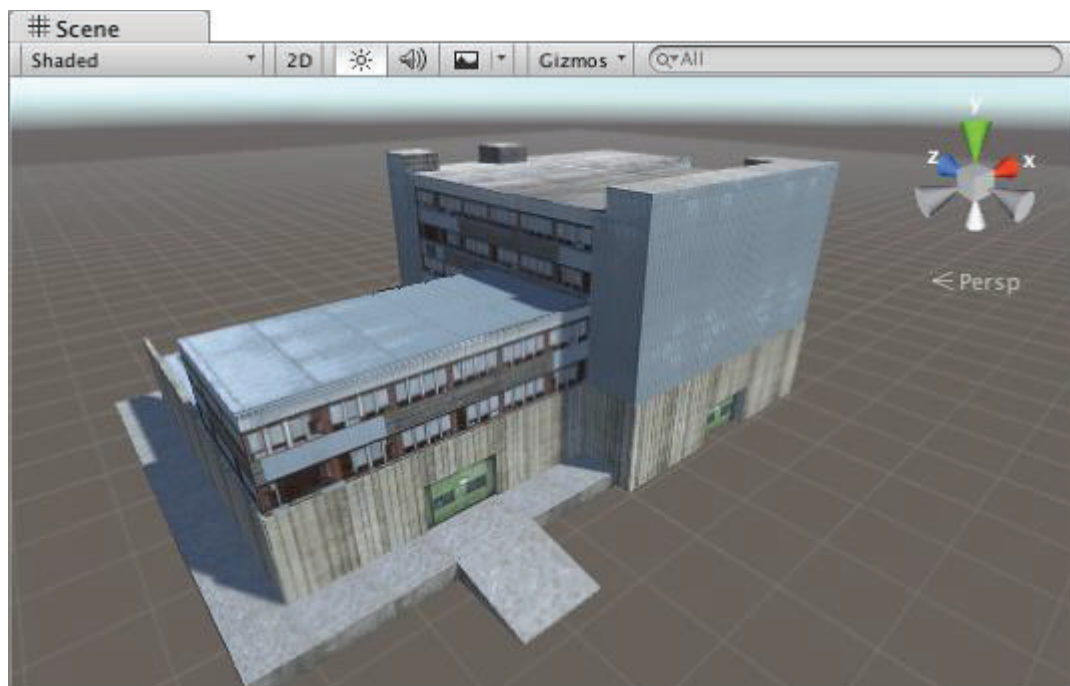


Ilustración 18: vista de la escena

- La **carpeta de proyecto** muestra todos los archivos relacionados con el proyecto y es la principal forma de navegar entre ellos. Entre los archivos relacionados se encuentran los modelos, los scripts o las animaciones, por ejemplo.



Ilustración 19: Carpeta de proyecto

- **Vista del juego** muestra el resultado final del juego, al pulsar el botón de play se compilarán todos los scripts del proyecto y mostrará como se vería el juego a ojos del jugador y podrás usar los mismos controles que tendría el jugador. En caso de error no avisarán por la consola.

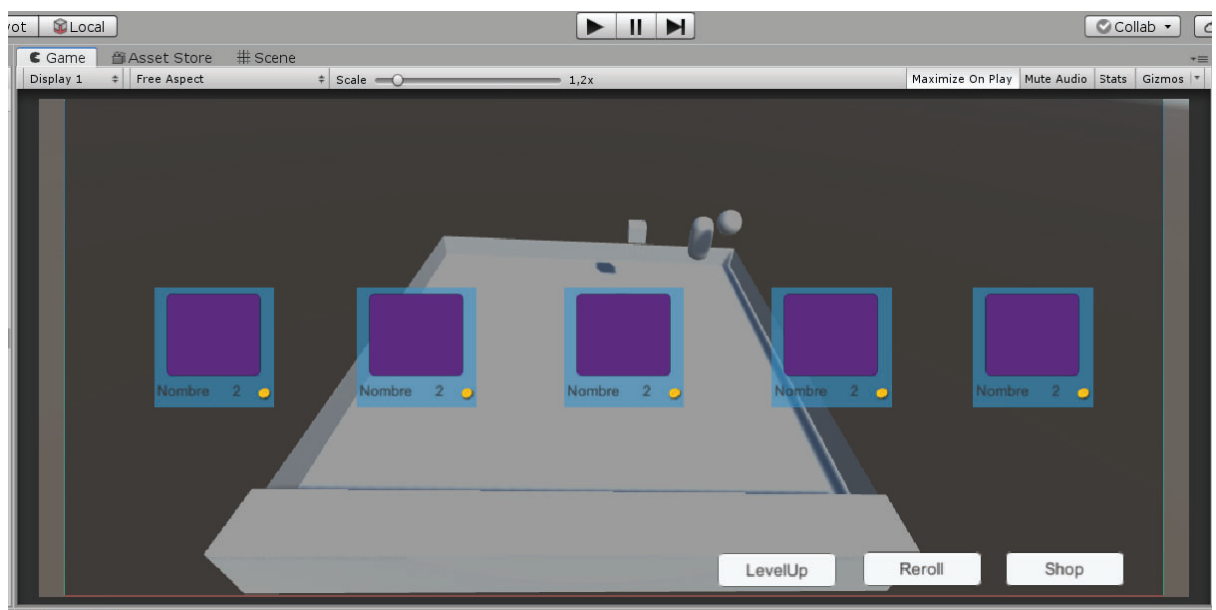


Ilustración 20: Vista del juego

- El **inspector** muestra la información del objeto seleccionado, te permite ver todas las propiedades como los materiales, componentes o los scripts que tenga. Desde aquí editaremos toda la información del objeto.

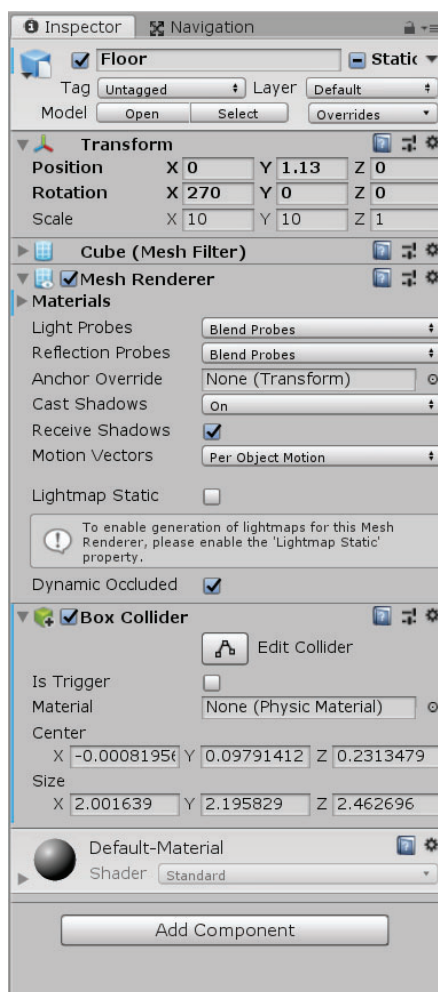


Ilustración 21: Inspector

- En la **vista de la consola** se mostrarán todos los mensajes de debugs y prints de los scripts además de los errores de compilación del proyecto.

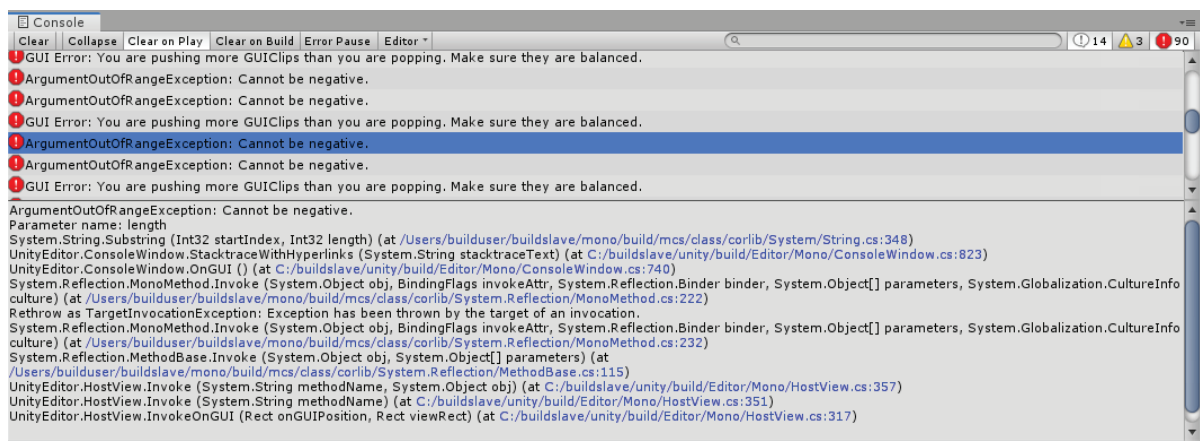


Ilustración 22: Vista de la consola

El motivo por el que me he decidido a usar este motor de videojuego como ya hemos mencionado antes es por su sencilla curva de aprendizaje y su gran comunidad que puedes usar para aprender, pero no solo por esto, también permite usar varios lenguajes de programación, puedes crear juegos para cualquier plataforma y tiene una gran cantidad de objetos prefabricados en caso de que no puedas hacer todos los elementos que necesite tu juego.

5.1.2 Visual Studio

El entorno de desarrollo integrado será Visual Studio 2017, es compatible con una multitud de lenguajes como C++, Java, .NET o C#, además también tiene todas las funcionalidades de Windows Azure.



Ilustración 23: Visual Studio

El lenguaje que utilizaremos para programar los scripts del juego será C# una de las opciones que ofrece Unity. C# es un lenguaje de programación multiparadigma, que abarca tipado fuerte, programación imperativa, programación declarativa, programación funcional y programación orientada a objetos, desarrollado y estandarizado por la empresa Microsoft.



Ilustración 24: C#

5.1.3 Blender

Blender es una aplicación multiplataforma que se puede ejecutar en Windows, Mac os y Linux, este programa está dedicado al modelado, iluminación, renderizado, animación y creación de gráficos tridimensionales. Actualmente es una aplicación de código abierto.

Blender tiene características avanzadas como despliegue UV, texturas, animación tridimensional con rigging, mezcla y formas. También es posible realizar todo tipo de simulaciones físicas de cuerpos para conseguir mejores efectos tridimensionales.

Este programa tiene una complejidad alta debido a la gran cantidad de tecnicismos que posee el programa y el amplio abanico de opciones que tiene.

Para hacer un modelo Blender tiene definidas distintas pantallas en las que realizar las tareas necesarias para el modelo que son:

- **Layout:** En esta pantalla podrás añadir modelos prefabricados a la escena y modificas su tamaño, posición y orientación.

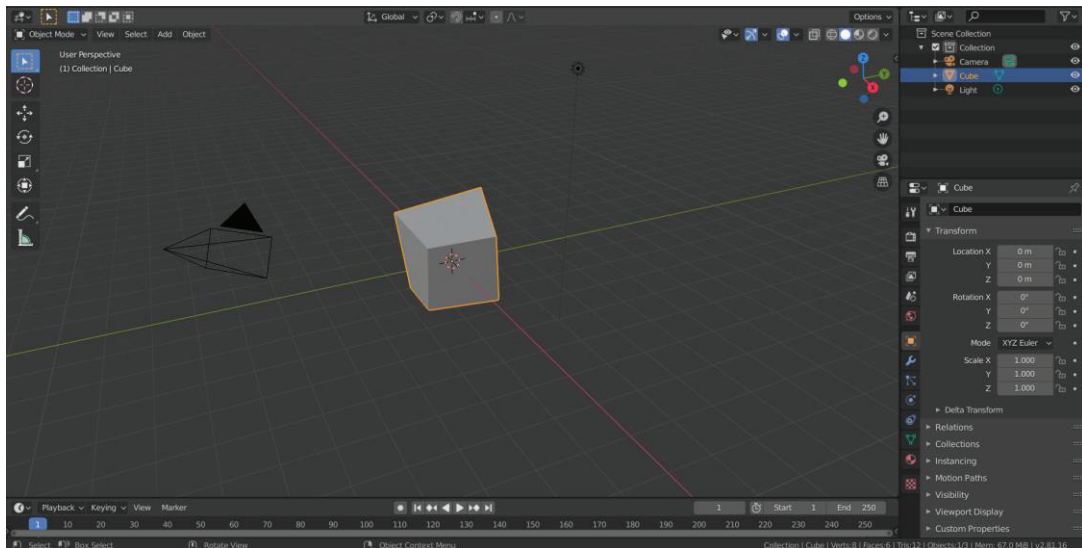


Ilustración 25: Layout

- **Modelado:** Aquí podrás modificar los objetos añadidos anteriormente. Podrás añadirles caras nuevas, modificar las aristas y vértices a tu gusto o incluso cortar el modelo por la parte que desees.

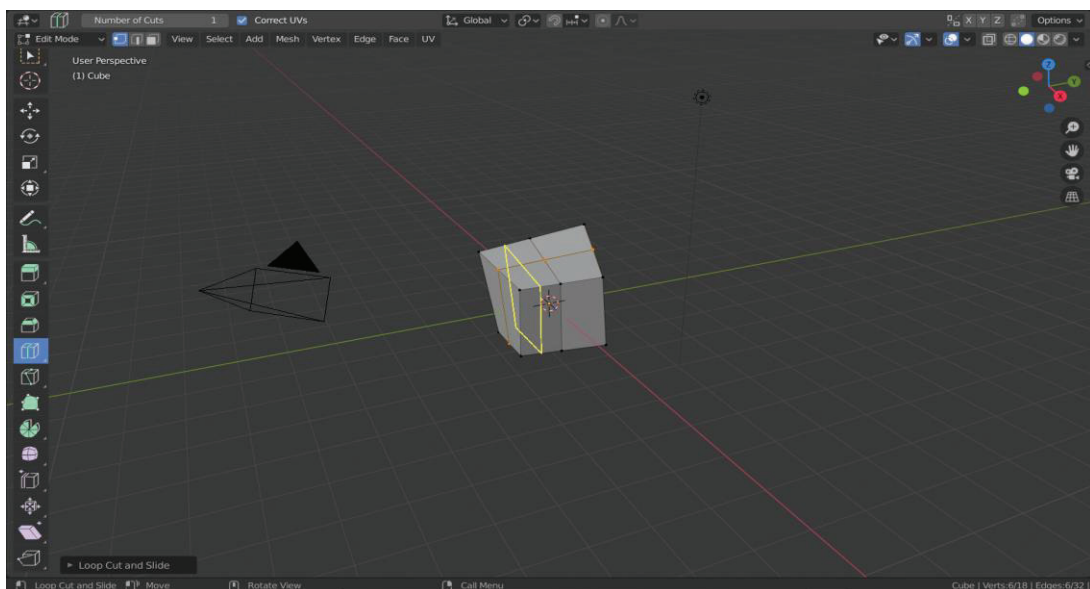


Ilustración 26: Modelado

- **Esculpir:** Esta pantalla nos permitirá esculpir nuestro modelo con distintas herramientas. La principal diferencia entre esta pantalla y la de modelado es que mientras modelado afecta a elementos individuales del modelo como los vértices en la pantalla de esculpir puedes modificar áreas enteras del modelo utilizando los diferentes cepillos que te ofrecen.

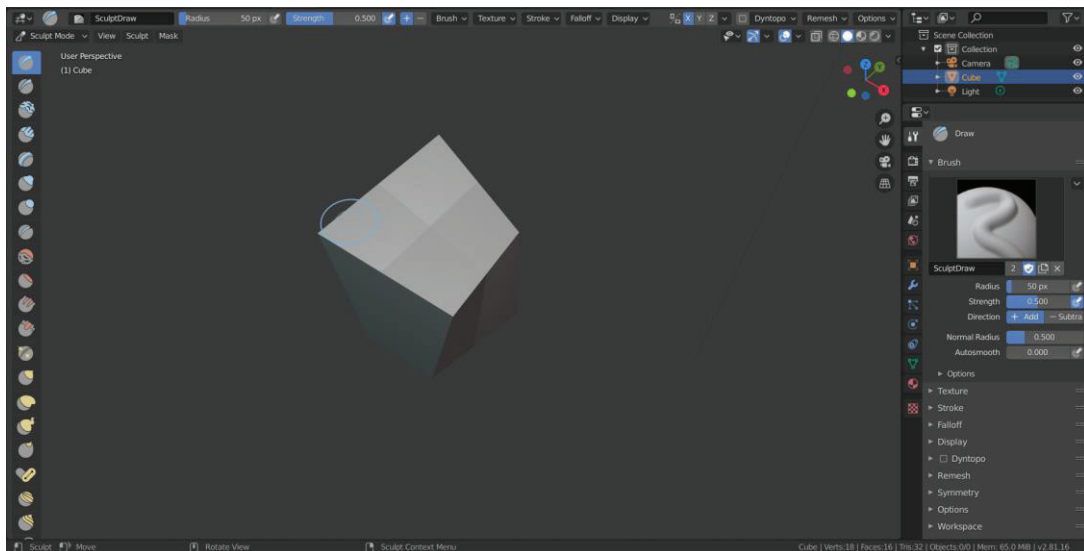


Ilustración 27: Esculpir

- **Edición UV:** Nos mostrara el modelo tridimensional como si fuera un desplegable en dos dimensiones. Sirve para detallar todas las caras a la vez y darle más realismo al modelo.

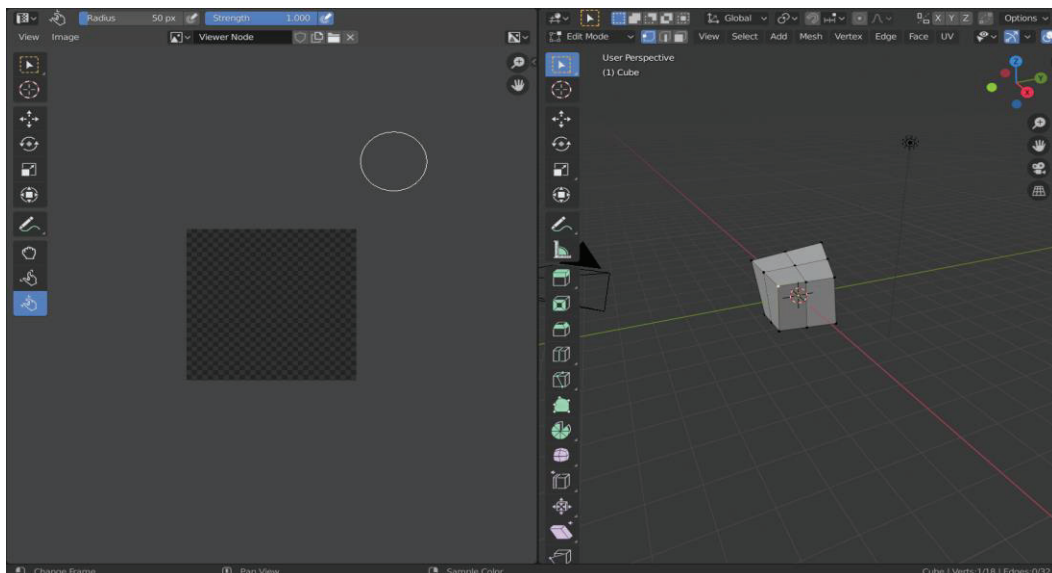


Ilustración 28: Edición UV

- **Pintar las texturas:** Aquí pintaremos el modelo y le daremos las texturas deseadas. Se puede pintar directamente sobre el modelo o sobre las caras que hemos desmontado en la anterior pestaña.

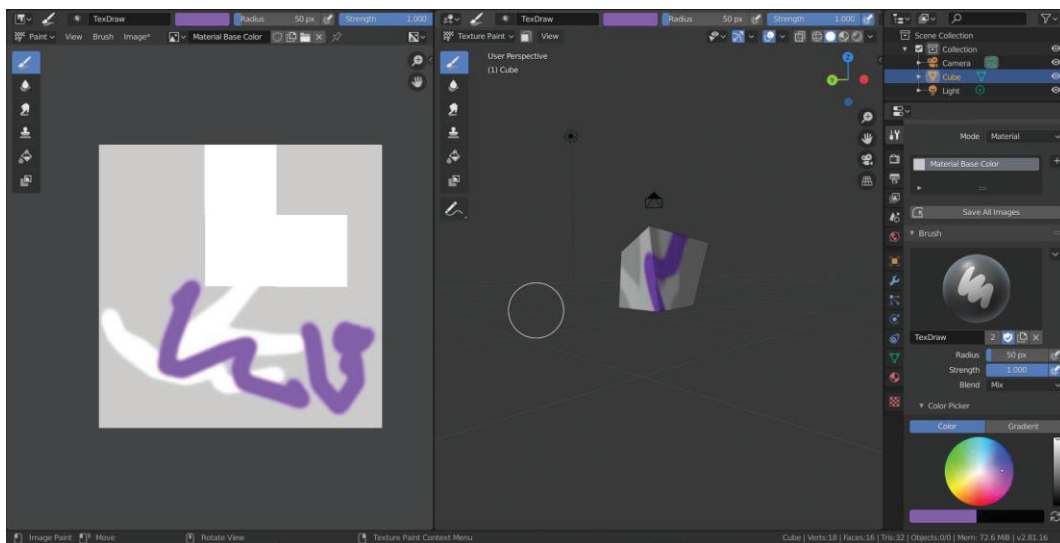


Ilustración 29: Pintar las texturas

- **Sombreado:** Sirve para darle sombras al modelo. Puedes modificar la cantidad de luz que recibe, como reflejan la luz o si son translucidos.

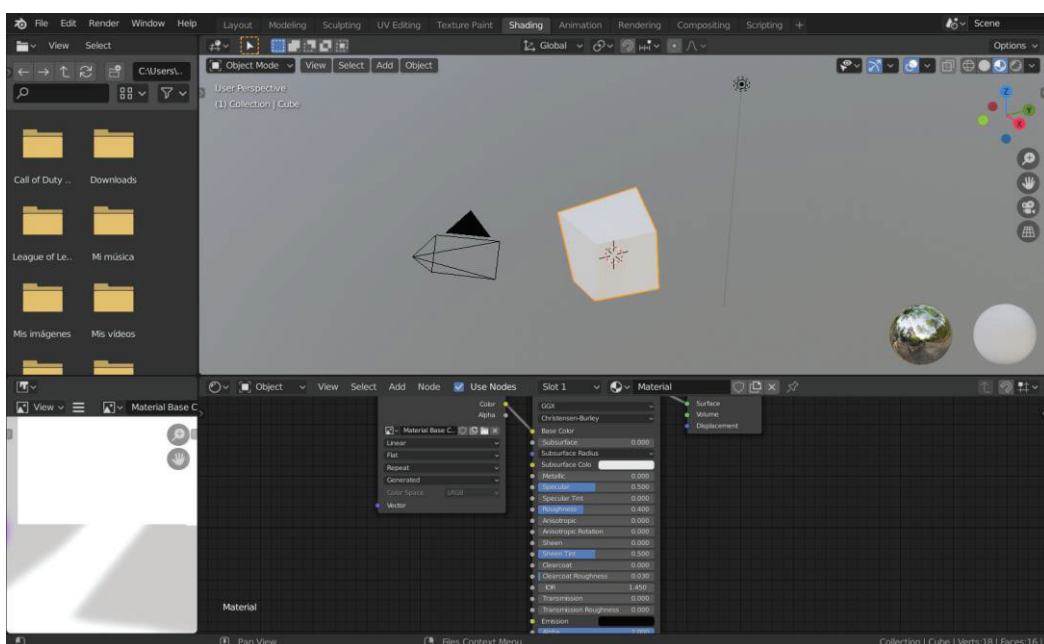
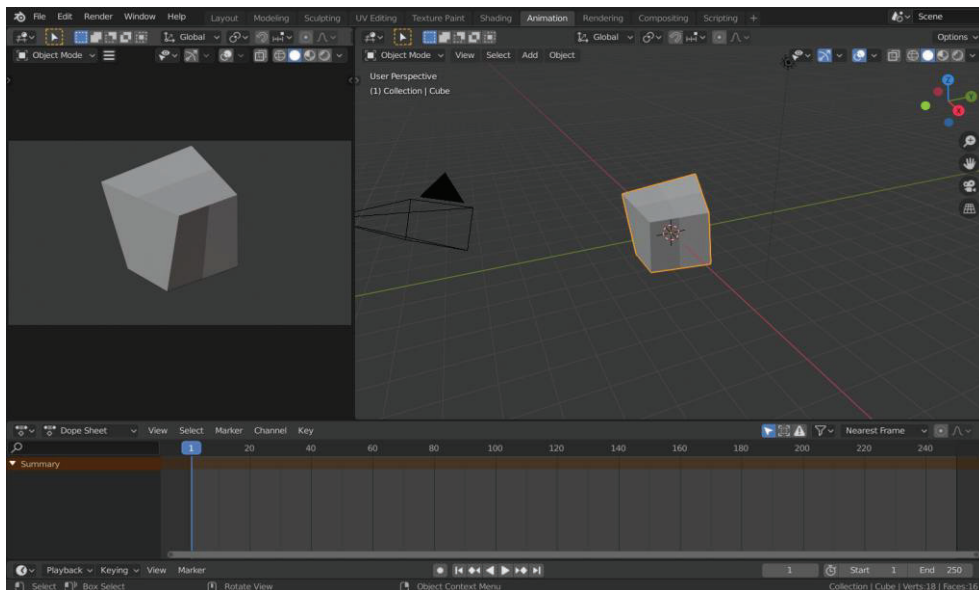
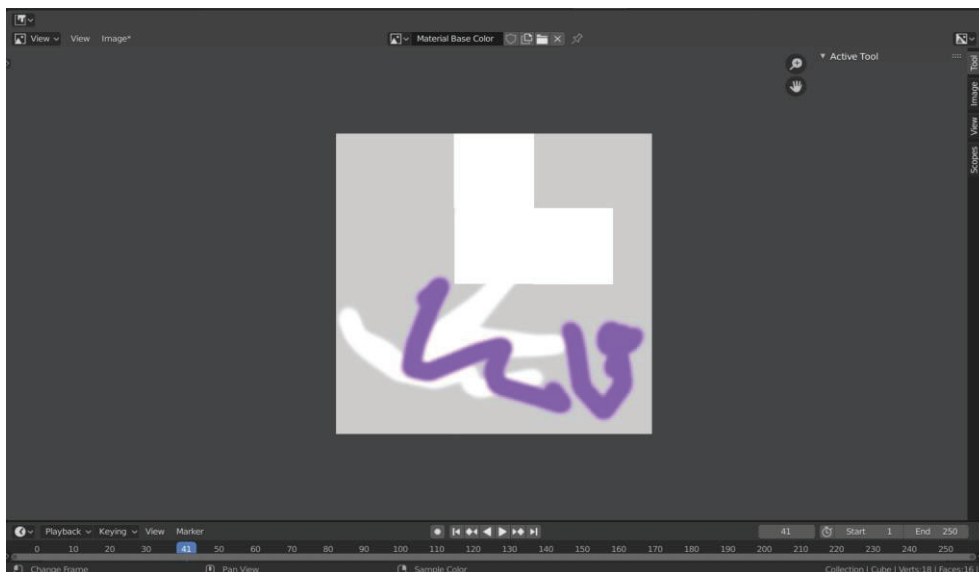


Ilustración 30: Sombreado

- **Animación:** Daremos movimiento a nuestro modelo. Ofrece herramientas para darle esqueletos al modelo y que los movimientos sean más naturales.

*Ilustración 31: Animación*

- **Renderizado:** Renderizado del modelo.

*Ilustración 32: Renderizado*

También tiene una pantalla en la que puedes escribir scripts de Python para automatizar ciertos procesos o hacerlos a través de código en vez de utilizar la interfaz.

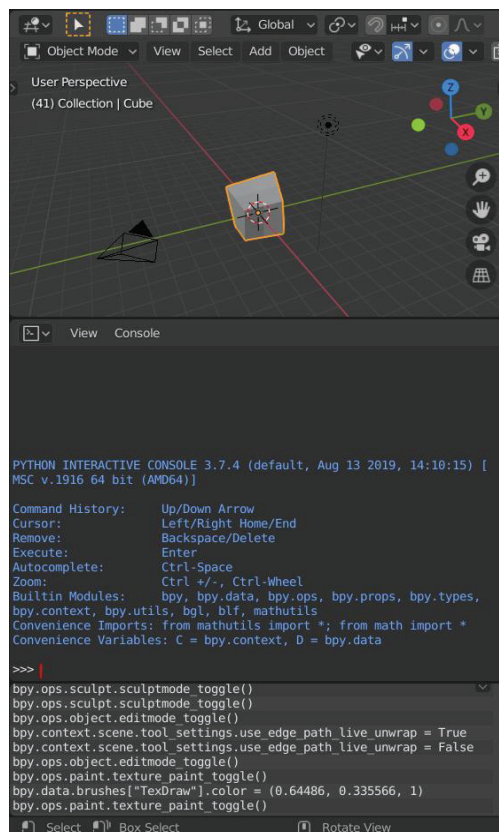


Ilustración 33: Pantalla de scripting

5.2 Gameplay

El juego va a pertenecer a al género de los Auto Battle pero va a recibir algunas modificaciones, la más llamativa es que no va a ser multijugador, se jugara contra la máquina que tendrá unidades aleatorias que a medida que avancen las rondas serán más fuertes.

El jugador tendrá un nivel y una barra de experiencia que indica lo que falta para subir de nivel. Para subir de nivel hay que rellenar la barra hay dos formas, la primera es pagando oro para subirla y la segunda esperar a que termine la ronda y de forma automática recibirás algunos puntos para la barra de experiencia. También tendrá un contador de vida que irá bajando a medida que pierda rondas.

También habrá oro como hemos mencionado antes que se ganará al final de cada ronda, aparte de comprar experiencia lo podemos usar para comprar unidades y refrescar la tienda.

Habrà nueve unidades, cada unidad tendrá un coste que abarcara el rango de uno de oro hasta 3 de oro, una sinergia, las opciones son guerrero, asesino y mago, y tendrán unas estadísticas determinadas. Las principales estadísticas son vida, ataque, rango que ataque y velocidad de ataque.

Al juntar tres unidades iguales se fusionarán creando la misma unidad de un nivel superior con la misma sinergia, pero con estadísticas mejoradas y aumento de coste. Las unidades podrán llegar hasta el nivel 3.

Las sinergias otorgarán bonificaciones los guerreros obtendrán más vida, los asesinos conseguirán más velocidad de ataque y los magos recibirán un aumento de daño. Para recibir la bonificación de la sinergia será necesario tener dos unidades distintas en el campo de batalla con la misma sinergia.

La tienda tendrá cinco unidades seleccionadas aleatoriamente al principio de cada ronda, la probabilidad de cada unidad dependerá del nivel del jugador, a más nivel mayor probabilidad de que salgan las unidades de mayos coste. La tienda también te permitirá refrescar la tienda para que salgan nuevas unidades por un coste de oro. Al comprar una unidad esta se añade automáticamente a tu banquillo y desaparece de la tienda.

La cantidad de unidades que puedes tener en el campo de batalla a la vez está limitada por el nivel del jugador, a más nivel más unidades en el campo puedes tener. Las unidades las puedes colocar como quieras para obtener mejoras estratégicas como proteger a tus unidades que hacen más daño.

5.3 Proceso de desarrollo del juego

Lo primero que se hizo fue concebir como se iba a jugar, teniendo en cuenta que el juego iba a pertenecer al género auto battle las bases estaban claras, el jugador tendría nivel, vida, unidades y una tienda donde comprar las unidades.

Debido a que esta va a ser la primera experiencia desarrollando un videojuego se decidió por eliminar algunas características claves del género como los objetos o el multijugador.

Una vez decidido como iba a ser el juego tocaba separarlo en dos fases principales, la primera seria elaborar los modelos que se iban a usar y la segunda desarrollar la lógica del juego.

Para la primera fase se utilizó la herramienta Blender en ella haríamos el campo de batalla que tendría un terreno de combate donde colocar las unidades activas y un banquillo donde guardar las unidades de reserva.

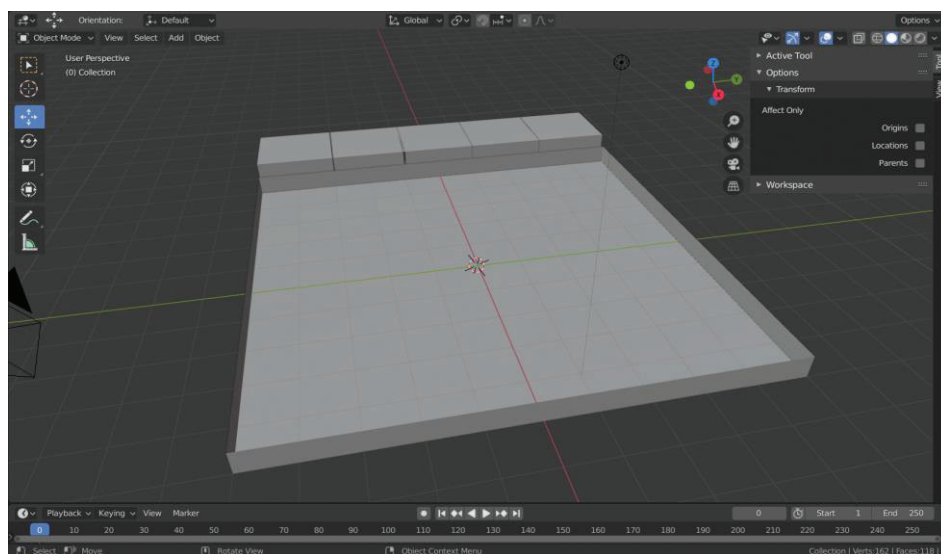


Ilustración 34: Imagen de campo de batalla

La unidad se decidió darles formas humanoides sencillas y para diferenciarlas cuando suben de nivel se les cambiara el color de algunos detalles. Los colores decididos para los detalles serán bronce, plata y oro ya que son colores que se diferencia bastante y es fácil asociarlos a la calidad de la unidad teniendo en cuenta como bronce la peor y oro la mejor.

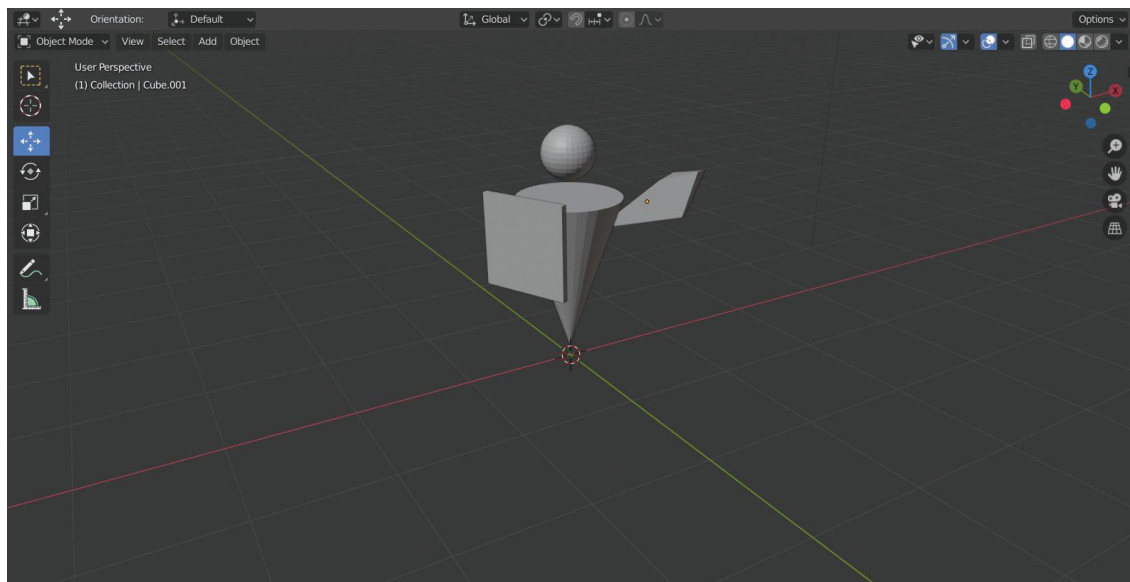


Ilustración 35: Concepto de unidad

Para la segunda fase se trabajó en Unity, esta fase se dividió en cinco grupos principales.

El primer grupo del que vamos a hablar es el de las unidades, se importaron los modelos en la anterior fase en Blender y los añadimos al proyecto como objetos del juego. Luego procederemos a añadirle las propiedades que le faltan.

Las propiedades más importantes que hay que añadir a las unidades son una caja de colisión para que las unidades tengan un cuerpo físico de verdad y no se superpongan y además le dará una zona de acción que se utilizara para saber cuándo las unidades están al alcance de las otras para atacarse. Otra propiedad muy importante es el Navmesh agent con el que la unidad podrá fijar el destino al que se quiere desplazar y como se va a desplazar, tanto la velocidad como la ruta que seguirá.

Por último, añadiremos los scripts de las unidades, en el script asignaremos sus estadísticas bases y sus estadísticas actuales en caso de que puedan ser modificadas más adelante, como la vida, el ataque o la velocidad de ataque. También gestionara toda la lógica de la unidad como por ejemplo si la unidad está en activo o está en el banquillo, si está muerta o viva, el nivel de unidad que es o buscara cual es el enemigo más cercano y lo fijara como destino hasta estar lo suficientemente cerca para realizar la acción de atacar e infligir daño a la unidad enemiga.

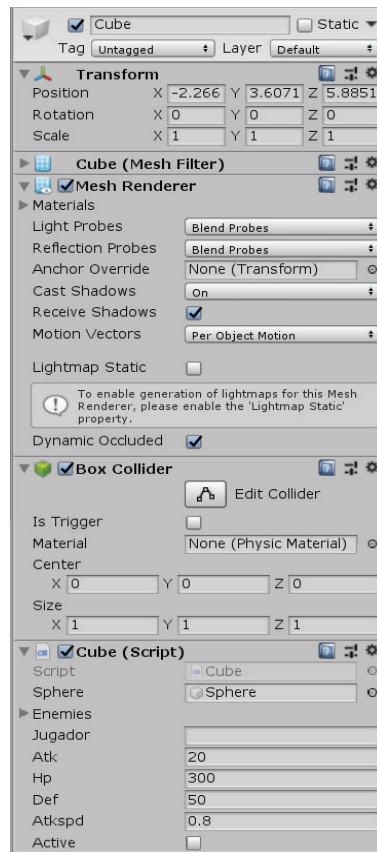


Ilustración 36: Algunos componentes de la unidad

Para el grupo de las interfaces utilizamos las herramientas UI que proporciona Unity, con ellas hicimos los menús y la interfaz de usuario que mostrar la ronda, la vida del jugador, el oro, el nivel y el tiempo restante que falta para que termine la ronda.

Los objetos más importantes para crear una interfaz son el canvas, dentro de él se colocarán todos los elementos de la interfaz, los paneles que se utilizarán para subdividir la interfaz y tenerlo más organizado y por últimos los botones y el texto.

En el grupo del gestor del juego añadiremos el modelo del campo de batalla y crearemos un script para que gestione la lógica del juego.

Al modelo se le pasará por una herramienta de Unity que le hará un renderizado de malla para poder establecer que partes del campo servirán de terreno para que las unidades se puedan mover y que parte se podrán colocar. También se le añadirá una caja de colisiones para poder conocer que unidades están en contacto con el terreno.

El script del gestor del juego se añadirá como componente a un objeto que su uso exclusivo será para ese script y que desde el script pueda acceder a todas las componentes del juego. El script gestionara las rondas, cuando empiezan y cuando acaban, cuanto daño recibe el jugador, cuando tiene que actualizarse la tienda o cuando y cuanto oro tiene que ganar el jugador. En resumen, se encargará de los aspectos generales del juego que involucren a varias partes.

La tienda se podría decir que es un subgrupo de la interfaz, pero al tener funciones y ser necesario interactuar con ella para desarrollar la experiencia del juego de forma adecuada. En la tienda utilizaremos las herramientas que proporciona Unity para la interfaz y añadiremos los scripts necesarios para que realicen las tareas para las que son diseñadas.

Las funciones que tendrá la tienda será comprar unidades que se añadirán de forma automática al banquillo del jugador si hay espacio, podrás actualizar la tienda en caso de que no quieras comprar ninguna unidad. Las unidades se actualizan al principio de cada ronda de forma aleatoria, los porcentajes de las unidades son el mismo para cada grupo de costes, cuanto menos nivel más probabilidad de que aparezcan las unidades de costes más bajos y viceversa cuanto más nivel más probable que aparezcan las unidades con mayor coste.

En la tienda aparecerán cinco unidades a la vez, cada unidad mostrara su coste de oro su nombre y una imagen suya para poder diferenciarlas de las otras.

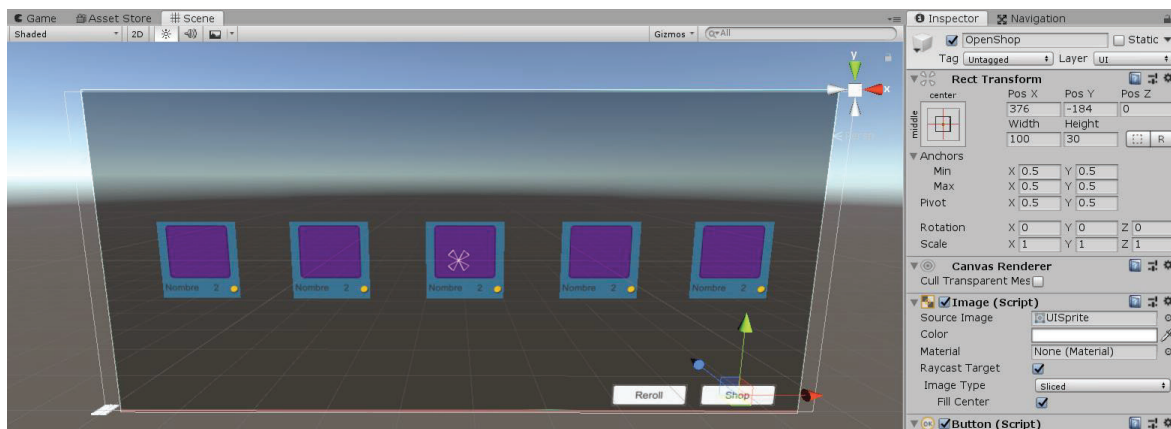


Ilustración 37: Panel de la tienda con alguno de sus componentes

Por último, el grupo del jugador, en él se creará un objeto del juego al cual le añadiremos las unidades que compre y le asignaremos un script para que gestione todas sus acciones y sus estadísticas.

El script gestionara la vida del jugador, cuanto tiene que bajar y pasara el valor a la interfaz para que se actualice, también gestionara el oro que tiene, cuanto gasta y si tiene suficiente para comprar, tendrá el valor del nivel del jugador y cuantas unidades podrá tener en el terreno de combate a la vez, además pasara el valor del nivel a la tienda para que pueda obtener las probabilidades adecuadas de aparición de las unidades, calculara cuanto le falta para subir de nivel y cuánto gana cada vez que termine la ronda o el jugador compre experiencia con oro.

6 Conclusiones y líneas futuras

En este capítulo hablaremos de las conclusiones obtenidas al realizar este trabajo, de las complicaciones que han ido surgiendo y mejor y cambios que se podrían realizar en el futuro para mejorar el juego.

6.1 Principales dificultades encontradas

La primera dificultad que me encontré fue aprender a usar Blender, un software muy complejo y con una gran cantidad de herramientas. En este programa aprender a hacer un simple modelo cuando no tienes mucho conocimiento puede ser muy complicado a pesar de la gran cantidad de guías y documentación que hay en internet y foros. En resumen, es una herramienta muy completa con una curva de aprendizaje muy alta y cuesta mucho iniciarse en el uso de esta herramienta.

Hacer un juego desde cero requiere un esfuerzo mucho mayor del estimado, los modelos como he hablado antes son mucho más complejos de lo que uno pueda pensar. Realizar la lógica del juego es diferente a la lógica de un programa y habrá que usar un enfoque parcialmente distinto. Además, hay que hacer muchas pruebas para asegurarse que todo funciona correctamente.

6.2 Mejoras y líneas futuras

La primera mejora que me gustaría añadir sería una mejora de los modelos, debido a que no soy un usuario experimentado de Blender los modelos acabaron siendo muy simples y toscos.

También me gustaría implementar la modalidad multijugador, en la que distintos jugadores puedan enfrentarse entre ellos.

Otra mejora sería añadir los objetos para las unidades, que se conseguirían de forma al azar matando unidades enemigas. Los objetos se equiparían a las unidades para que obtengan distintas bonificaciones.

Y por último se podrían añadir mas unidades y sinergias para dar más posibilidades de combinaciones de unidades y sinergias. Añadiendo esto obtendríamos una experiencia de juego mucho más variada.

6.3 Conclusiones

Mi objetivo era aprender todo el proceso que hay detrás del desarrollo de un videojuego e iniciarme en el sector. La conclusión principal que saco es que desarrollar un videojuego es mucho mas complejo de lo que pueda parecer a simple vista. Aunque algunos programas son mas amigables para usuarios novatos hay otros que se necesita una gran experiencia hasta empezar a usarlos con soltura.

Este trabajo de fin grado me ha servido para iniciarme en el sector de los videojuegos y tener una idea del trabajo que conlleva el proceso de creación de videojuegos.

Para completar este proyecto se han requeridos esfuerzos adicionales a los estimados al inicio de la planificación, debido a una mala estimación por mi parte del esfuerzo que conlleva realizar un videojuego.

7 Bibliografía

- [1] S. Belli y C. López Raventós, «Breve historia de los videojuegos,» *Athenea Digital*, n° 14, pp. 159-179, 2013.
- [2] L. Sacristán, «Xataka,» 10 Mayo 2020. [En línea]. Available: <https://www.xatakandroid.com/aplicaciones-android/estos-juegos-android-descargados-historia-google-play>. [Último acceso: 5 Junio 2020].
- [3] J. Tones, «Xataka,» 10 Enero 2020. [En línea]. Available: <https://www.xataka.com/videojuegos/pokemon-go-no-estaba-muerto-estabatiendo-record-ingresos-casi-900-millones-dolares-estimados-2019-superando-ano-su-debut>. [Último acceso: 21 Marzo 2020].
- [4] S. Fernández, «Xataka,» 28 Enero 2016. [En línea]. Available: <https://www.xatakamovil.com/aplicaciones/clash-of-clans-es-el-juego-movil-que-mas-ingresos-genera-y-a-mucha-distancia-del-segundo>. [Último acceso: 12 Enero 2020].
- [5] G. N. José, «Xataka,» 7 Julio 2019. [En línea]. Available: <https://www.xatakamovil.com/aplicaciones/auto-battlers-todo-genero-que-auto-chess-ha-puesto-moda-que-empieza-a-triunfar-moviles>.
- [6] A. González, «Vandal,» 30 Septiembre 2019. [En línea]. Available: <https://vandal.elespanol.com/noticia/1350727602/teamfight-tactics-tiene-33-millones-de-jugadores-mensuales/>. [Último acceso: 20 Febrero 2020].
- [7] TIC Portal, «TIC Portal,» 11 Septiembre 2018. [En línea]. Available: <https://www.ticportal.es/glosario-tic/gestion-proyectos>. [Último acceso: 20 junio 2020].
- [8] R. G. Figueroa, C. J. Solis y A. A. Cabrera, «METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES,» 2008.
- [9] D. Avedaño, «Workep,» 18 Agosto 2018. [En línea]. Available: <https://blog.workep.com/es/metodologias-de-gestion-de-proyectos-tradicional-vs-agil>. [Último acceso: 24 Junio 2020].
- [10] L. M. Larriba, «Billage,» 22 Junio 2016. [En línea]. Available: <https://www.getbillage.com/es/blog/metodologia-kanban-ventajas-y-caracteristicas>. [Último acceso: 19 Diciembre 2019].