

Prototipo de videojuego para PC compatible con mando de Xbox, clase *ARPG*, estilo *Cyberpunk*, con vistas isométricas

Autor: Mario Rodes Ruiz

Tutor: Joel Servitja Feu

Profesor: Joan Arnedo Moreno

Créditos/Copyright



Esta obra está sujeta a una licencia de Reconocimiento- NoComercial-SinObraDerivada

[3.0 España de Creative Commons.](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Descriptivo del trabajo</i>
Nombre del autor:	<i>Mario Rodes Ruiz</i>
Nombre del colaborador/a docente :	<i>Joel Servitja Feu</i>
Nombre del PRA:	<i>Joan Arnedo Moreno</i>
Fecha de entrega (mm/aaaa):	<i>01/2023</i>
Titulación o programa:	<i>Grado de Técnicas de Interacción Digital y Multimedia</i>
Área del Trabajo Final:	<i>Videojuegos</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Top-Down, Cyberpunk, ARPG</i>

Resumen del Trabajo (máximo 250 palabras):

Durante los últimos años, el auge de las TIC en el ámbito de la industria del desarrollo de los videojuegos ha dado como fruto plataformas como el *Asset Store* de *Unity* donde hoy en día se venden infinidad de herramientas, así como de medios capaces de simplificar las tareas de producción a costa de la estandarización y la falta de contenido exclusivo original.

La finalidad de este proyecto es la de crear un videojuego con entornos, objetos y personajes cuidados, ricos en detalles, por medio de un proceso creativo original. Para ello se hará uso de una apariencia que aúne la estética de estos en la búsqueda de un producto con una identidad visual con rasgos propios y distinguibles.

Este se ejecutará en ordenadores con sistemas operativos Windows que sean capaces de cubrir las necesidades de hardware que requerirá. Será compatible con el mando de Xbox y se tratará de un *ARPG* con estética *cyberpunk* y vistas isométricas que permita al jugador recrearse en el escenario.

Se gestionará con una metodología de trabajo ágil, concretamente la SCRUM. Permitirá dividir en *sprints* los distintos hitos que se llevaran a cabo durante el desarrollo del proyecto. Los cuales abarcan principalmente procesos creativos de diseño de imágenes, modelado y texturizado, así como su posterior montaje y programación en el motor gráfico.

A nivel funcional estará compuesto por un menú básico y un nivel jugable que permita al usuario hacerse una buena idea de cómo podría ser el juego con una historia y más niveles. Aun así, pese a que se trata de un juego nivel único, será capaz de ejecutar todas las funciones básicas que se esperan de un título terminado.

Como característica principal, se primará crear un juego con un toque más artístico, inmersivo y único que sea capaz de ofrecer aire fresco en un entorno (el de los videojuegos) donde cada vez es más común ver escenarios con objetos genéricos, incoherentes entre sí y con poca credibilidad.

Abstract (in English, 250 words or less):

In recent years, the rise of ICTs in the field of video game development has given rise to platforms such as the Unity Asset Store, where today countless tools are sold, as well as 3D models capable of simplifying the production process at the cost of standardization and the lack of original exclusive content.

The purpose of this project is to create a video game with environments, objects, and characters, rich in details, through an original creative process. For this, an appearance will be used that combines the aesthetics of these in the search for a product with a visual identity with its own and distinguishable features.

This will run on computers with Windows operating systems that can meet the hardware needs that it will require. It will be compatible with the Xbox controller and it will be an ARPG with cyberpunk aesthetics and isometric views that allow the player to recreate himself on stage.

It will be managed with an agile work methodology, specifically the SCRUM. This allows the different milestones that will be carried out during the development of the project to be divided into sprints. Which mainly cover creative processes of image design, modelling, and texturing, as well as their subsequent assembly and programming in the graphics engine.

At a functional level, it will be made up of a basic menu and a playable level that allows the user to get a good idea of what the game could be like with a story and more levels. Even so, despite the fact that it is a single-level game, it will be able to execute all the basic functions that are expected from a finished title.

As the main feature, the priority will be to create a game with a more artistic, immersive, and unique touch that can offer fresh air in an environment (that of video games) where it is increasingly common to see scenarios with generic objects, inconsistent with each other and with weak credibility.

Dedicatoria

Dedicado a todas y cada una de las personas que me quieren y siempre me han querido. A aquellas que desde que me conocieron nunca han dejado de quererme y nunca lo harán. A todas aquellas que han creído y confiado en mí incondicionalmente, frente a cualquier situación o dificultad.

A todos, gracias por haberme ayudado a llegar hasta aquí. Os quiero.

Abstract

Durante los últimos años, el auge de las TIC en el ámbito de la industria del desarrollo de los videojuegos ha dado como fruto plataformas como el *Asset Store* de *Unity* donde hoy en día se venden infinidad de herramientas, así como de medios capaces de simplificar las tareas de producción a costa de la estandarización y la falta de contenido exclusivo original.

La finalidad de este proyecto es la de crear un videojuego con entornos, objetos y personajes cuidados, ricos en detalles, por medio de un proceso creativo original. Para ello se hará uso de una apariencia que aúne la estética de estos en la búsqueda de un producto con una identidad visual con rasgos propios y distinguibles.

Este se ejecutará en ordenadores con sistemas operativos Windows que sean capaces de cubrir las necesidades de hardware que requerirá. Será compatible con el mando de Xbox y se tratará de un ARPG con estética *cyberpunk* y vistas isométricas que permita al jugador recrearse en el escenario.

Se gestionará con una metodología de trabajo ágil, concretamente la SCRUM. Permitirá dividir en *sprints* los distintos hitos que se llevaran a cabo durante el desarrollo del proyecto. Los cuales abarcan principalmente procesos creativos de diseño de imágenes, modelado y texturizado, así como su posterior montaje y programación en el motor gráfico.

A nivel funcional estará compuesto por un menú básico y un nivel jugable que permita al usuario hacerse una buena idea de cómo podría ser el juego con una historia y más niveles. Aun así, pese a que se trata de un juego nivel único, será capaz de ejecutar todas las funciones básicas que se esperan de un título terminado.

Como característica principal, se primará crear un juego con un toque más artístico, inmersivo y único que sea capaz de ofrecer aire fresco en un entorno (el de los videojuegos) donde cada vez es más común ver escenarios con objetos genéricos, incoherentes entre sí y con poca credibilidad.

Palabras clave

Top-Down, Cyberpunk, ARPG, modelling, videogame

Índice

1. Introducción.....	11
1.1. Introducción/Prefacio.....	11
1.1.1. Motivación personal	13
1.2. Descripción/Definición	15
1.3. Objetivos generales	16
1.3.1. Objetivos principales	16
1.3.2. Objetivos secundarios	16
1.4. Metodología y proceso de trabajo.....	17
1.5. Planificación.....	20
1.5.1. Diagrama de gantt.....	23
1.6. Presupuesto	23
1.7. Estructura del resto del documento	25
2. Análisis de mercado	26
2.1. Estado del mercado	26
2.2. Público objetivo y perfiles de usuario (Buyer persona)	27
2.2.1. Buyer personas.....	28
2.3. Análisis de la competencia	36
2.3.1. <i>The Ascent</i>	37
2.3.2. <i>Ruiner</i>	39
2.3.3. <i>I see red</i>	41
2.3.4. <i>Synthetik 2</i>	43
2.3.5. <i>Jydge</i>	45
2.4. Análisis DAFO.....	47
2.4.1. <i>The Ascent</i>	47
2.4.2. <i>Ruiner</i>	48
2.4.3. <i>I see red</i>	48
2.4.4. <i>Synthetik 2</i>	49
2.4.5. <i>Jydge</i>	49
2.5. Mapa conceptual (competitive landscape).....	50
3. Propuesta.....	51

3.1. Definición de objetivos/especificaciones del producto	51
3.2. Modelo de negocio	51
3.3. Estrategia de marketing.....	52
4. Diseño.....	54
4.1. Arquitectura general del juego	54
4.2. Arquitectura detallada del videojuego y sus funciones	55
4.2.1. Arquitectura interna del juego.....	55
4.2.2. Navegación.....	61
4.2.3. Contenidos.....	62
4.3. Diseño gráfico, de interfaces y de jugabilidad	62
4.3.1. Estilos	62
4.3.2. Jugabilidad (<i>Twin-stick shooter</i>).....	63
4.4. Motor gráfico y assets. DDCs e IDEs	64
4.5. Diseño del nivel	66
4.5.1. Idea original	66
4.5.2. Primeros bocetos.....	67
4.5.3. Modelos incorporados de creación propia.....	69
4.5.4. Modelos y texturas usados de terceros	73
4.5.5. Elementos interactivos.....	76
5. Implementación.....	78
5.1. Requisitos técnicos recomendados	78
5.2. Instrucciones de ejecución	78
6. Demostración	79
6.1. Ejemplo de uso del producto o guía de usuario	79
7. Conclusiones y líneas de futuro	81
7.1. Conclusiones	81
7.2. Líneas de futuro.....	82
Bibliografía.....	83
Anexos	85

Figuras y tablas

Lista de imágenes, tablas, gráficos, diagramas, etc., numeradas, con títulos y las páginas en las cuales aparecen.

Índice de figuras

Figura 1: Estadística de ingresos globales del sector videojuegos de la página <i>www.pwc.com</i>	11	
Figura 2: <i>CastleMiner Z</i> . Una copia más del <i>Minecraft</i> producto de un collage de <i>assets</i>	12	
Figura 3: <i>MASS Extinction</i> . Antiguo proyecto. 2022.....	13	
Figura 4: <i>The Ascent</i> . Nominado al juego "Outstanding Visual Style" en el <i>Steam Awards</i> del año 2021.....	14	
Figura 5: <i>Diagrama de gantt</i>	23	
Figura 6: <i>The Ascent</i> . Nominado al juego "Outstanding Visual Style" en el <i>Steam Awards</i> del año 2021. Fuente: <i>Steam</i>	37	
Figura 7: <i>Ruiner (2017)</i> . Fuente: <i>Steam</i>	39	
Figura 8: <i>I see red (2022)</i> . Fuente: <i>Steam</i>	41	
Figura 9: <i>Synthetik 2 (2021)</i> . Fuente: <i>Steam</i>	43	
Figura 10: <i>Jydge (2017)</i> . Fuente: <i>Steam</i>	45	
Figura 11: Mapa conceptual (<i>competitive landscape</i>).....	50	
Figura 12: Partes principales de un videojuego (según Unity Learn).....	54	
Figura 13: <i>Diagrama de los componentes del juego, relación y jerarquía</i>	56	
Figura 14: Menús y navegación del juego.....	61	
Figura 15: Infografía controles del prototipo.....	63	
Figura 16: Inventario junto con los botones con etiquetas identificativas y colores.....	64	
Figura 17: Nivel visto desde arriba en Unity.....	66	
Figura 18: Boceto del Nivel 1.....	67	
Figura 19: Boceto de pilar de materiales.....	68	
Figura 20: Boceto de la mesa holográfica.....	68	
Figura 21: Nave espacial.....	69	
Figura 22: Peana para la nave espacial con rampa.....	69	
Figura 23: Pilar del boceto de la figura 19 modelado y texturizado.....	70	
Figura 24: Pilares dentro del nivel exponiendo materiales extraños.....	70	
Figura 25: Mesa del boceto de la figura 20 modelado y texturizado.....	71	
Figura 26: Mesa de la figura anterior dentro del nivel en Unity con un mapa topográfico encima.....	71	
Figura 27: Peana solar.....	72	
Figura 28: Peana solar dentro del nivel con los planetas y sus texturas descargados de <i>Adobe Stock</i>	72	
Figura 29: Cajas de armas y municiones.....	73	
Figura 30: Esferas de vida.....	73	
Figura 31: Enemigo de 50 puntos de vida	Figura 32: Enemigo de 200.....	73
Figura 33: Personaje jugable	Figura 34: Enemigo/jefe con 1000 puntos de vida.....	74
Figura 35: Mesas, altavoces sillones, portátiles y pantallas del <i>asset "Top Down Engine"</i>	74	
Figura 36: Muebles modificados del <i>asset "Top Down Engine"</i> con materiales luminosos y metálicos.....	75	
Figura 37: Muebles modificados del <i>asset "Top Down Engine"</i> con un material emisor con texto luminoso.....	75	

Figura 38: Cubos destruibles con objetos aleatorios.....	76
Figura 39: Puertas (destruibles).....	77
Figura 40: Pilares (destruibles).....	77
Figura 41: Archivos de código fuente y ejecutables.....	78

Índice de tablas

Tabla 1: Recursos (<i>software, assets, plugins</i>).....	19
Tabla 2: Métodos o técnicas.....	19
Tabla 3: Planificación PEC1.....	20
Tabla 4: Planificación PEC2.....	21
Tabla 5: Planificación PEC3.....	21
Tabla 6: Planificación PEC4.....	22
Tabla 7: Planificación PEC5.....	22
Tabla 8: Costos derivados de la adquisición de <i>software, assets y plugins</i>	24
Tabla 9: Costes derivados de los recursos humanos.....	24
Tabla 10: Buyer person 1 - David.....	28
Tabla 11: Buyer person 2 - Emily.....	29
Tabla 12: Buyer person 3 - Ulrich.....	30
Tabla 13: Buyer person 4 - Buffy.....	31
Tabla 14: Buyer person 5 - Joe.....	32
Tabla 15: Buyer person 6 - Gabriella.....	33
Tabla 16: Buyer person 7 - Pierre.....	34
Tabla 17: Buyer person 8 - Lucía.....	35
Tabla 18: DAFO <i>The Ascent</i>	47
Tabla 19: DAFO <i>Ruiner</i>	48
Tabla 20: DAFO <i>I see red</i>	48
Tabla 21: DAFO <i>Synthetik 2</i>	49
Tabla 22: DAFO <i>Jydge</i>	49
Tabla 23: DDCs e IDEs usados en el proyecto y la razón.....	65

1.Introducción

En esta sección se tratan las motivaciones que han impulsado el proyecto, sus características el objetivo que se quiere alcanzar, la planificación detallada de su desarrollo, así como la metodología de trabajo que se empleará para llevarlo a cabo.

1.1. Introducción/Prefacio

El aumento del uso de los videojuegos, así como la cantidad de ingresos que generan cada año a nivel mundial parece no tener freno. Muy al contrario, la industria de los videojuegos está en auge y experimenta un aumento vertiginoso. Como referencia, en 2026 según el nuevo informe del PwC (*PriceWaterhouseCoopers*), el “*Perspectives from the Global Entertainment & Media Outlook 2022–2026*”, se espera que la industria del videojuego alcance los 326 mil millones de dólares, en comparación a los 235 mil millones de dólares que se esperan a final de este año, casi un 30 % más que en la actualidad.

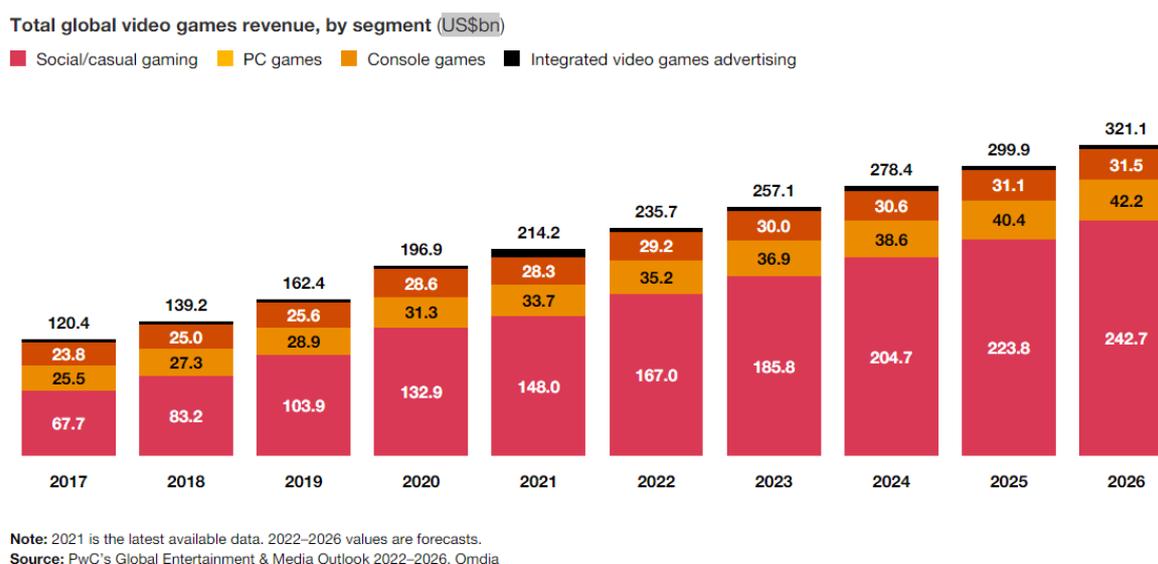


Figura 1: Estadística de ingresos globales del sector videojuegos de la página www.pwc.com

El aumento en el uso de los videojuegos, sobre todo los del estilo en línea multijugador, ha sido posible en buena parte gracias al rápido desarrollo y evolución de las TIC (Tecnologías de la información y la comunicación). Estas mejoran de año en año proporcionando tanto hardware como softwares nuevos. Mas potentes y sofisticados que mejoran la calidad y potencia de los videojuegos hasta límites cada vez más increíbles.

Este incremento tanto en el uso de los videojuegos, como de las tecnologías que le sirven de base, también traen consigo un cambio en los procesos de desarrollo de los videojuegos, así como sus metodologías de trabajo y herramientas de trabajo. Si bien un juego antiguamente se debía de desarrollar prácticamente desde cero (como el *Unreal Tournament* o el primer *Doom*), hoy en día existen cantidad de empresas tercerizadas que se dedican a desarrollar *middelwares*, *plugins* y *assets* que ahorran trabajo y estandarizan procesos, entre ellos artísticos, ya que se venden entornos modelos y personajes.

Esto afecta de manera positiva a todos, pero también de manera negativa. Si bien existe la capacidad de crear videojuegos más rápidamente y de manera más sencilla, esto también conlleva el aumento de la cantidad de títulos que son muy simples y carecen de carácter e identidad propia. En el peor de los casos nos podemos encontrar un conjunto de *assets* que dan como resultado un producto inconsistente y vacío, incoherente y sin sentido, muchas veces con estéticas “collage” donde los elementos que conforman los entornos no tienen ninguna conexión.

El usuario puede acabar jugando títulos “collage” de mala calidad, o copias de títulos ya existentes, en los que lo único que realmente hace es perder tiempo y/o dinero (micro transacciones o comprar el juego) hasta que reacciona a la mala calidad del contenido, lo que le conduce a abandonarlo. Pese a todo, no todos los jugadores abandonan estos juegos. Como resultado, eclipsan la posibilidad a proyectos más consolidados, coherentes y elaborados de hacerse un hueco en mercado. Un hecho que, si no controlamos, puede desembocar en una pérdida de calidad de los títulos en un gran porcentaje del catálogo.



Figura 2: *CastleMiner Z*. Una copia más del *Minecraft* producto de un collage de *assets*

Por suerte, las nuevas tecnologías nos permiten elevar cada vez más el promedio de la calidad grafica de los videojuegos. Hecho que ofrece la oportunidad a los creativos de poder desarrollar juegos ricos en detalles y muy elaborados, en detracción de los monótonos, aburridos y vacíos niveles de los juegos “collage”, que poco invierten en crear una buena experiencia de juego que deje satisfecho al jugador.

1.1.1. Motivación personal

Si bien el antiguo proyecto que desarrollé se trataba de un videojuego de características similares a las de este, tenía ciertos hándicaps que lo ponían en desventaja. Se trataba de un juego de Android (falta de hardware para gráficos), con una estética semi simplista, aunque llamativa (rico en formas, pero texturas muy básicas) y modelos genéricos como personajes.



Figura 3: MASS Extinction. Antiguo proyecto. 2022

No contento del todo con el resultado, me he propuesto crear un proyecto en donde pueda corregir dichas desventajas desarrollando uno nuevo desde cero y encargándome personalmente de crear los objetos que aparezcan en escena. El resultado pretende asemejarse a títulos como “The Ascent”(nominado al juego “Outstanding Visual Style” en el Steam Awards del año 2021) en cuanto a nivel de detalles, calidad de modelos, luces y texturizados.

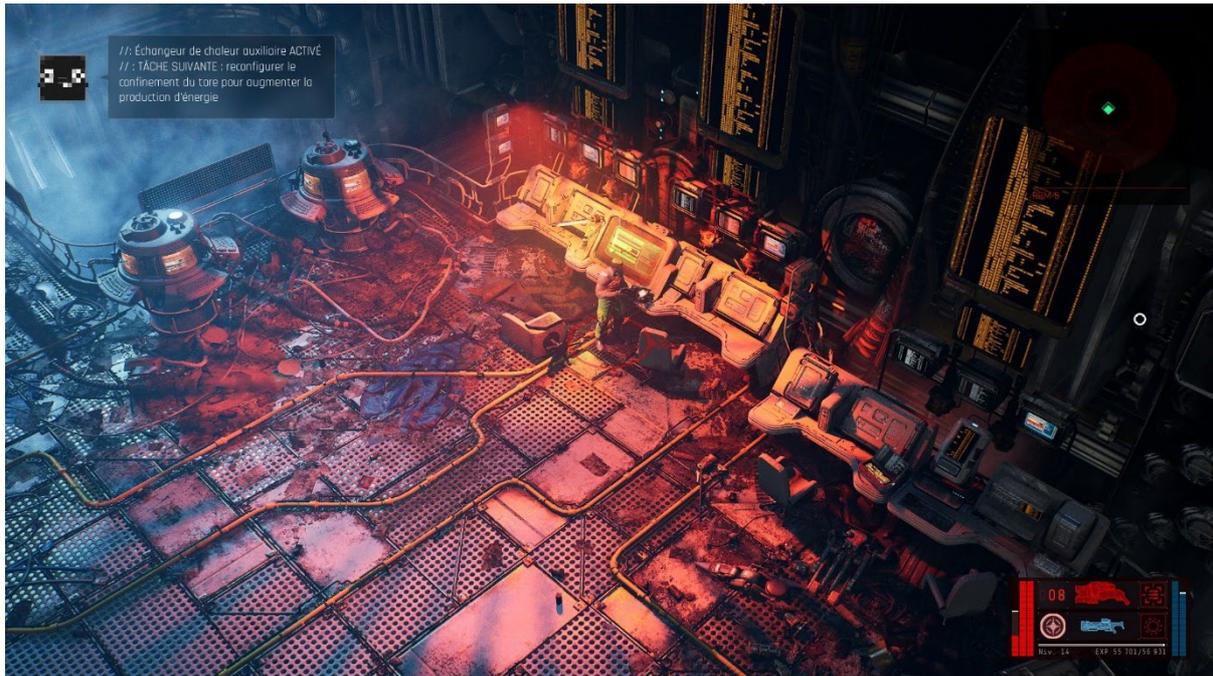


Figura 4: *The Ascent*. Nominado al juego "Outstanding Visual Style" en el *Steam Awards* del año 2021

Es cierto que igualar el nivel de calidad visual y técnico de "*The Ascent*" es una tarea imposible para mí solo con el periodo de tiempo del que dispongo, pero existen otros aspectos que si pueden conseguirse. Es posible crear una estética unificada y característica que defina la identidad del juego con la apariencia más realista posible (texturas, luces y formas) y así, evitar el uso de cualquier tipo de modelos en 3D genéricos que le resten carácter.

1.2. Descripción/Definición

Se pretende crear un videojuego con la ayuda del motor gráfico Unity, algún asset de scripts a nivel de gestor de funciones del juego, una librería de efectos de sonido y modelos 3D originales realizados para el juego en Blender.

Este se creará con la intención de cubrir la necesidad de juegos indie con estéticas originales en el mercado de PC que no utilicen recursos visuales comerciales para crear sus entornos y personajes. Hecho que resta variedad y genialidad a los títulos contemporáneos diseñados a través de estas metodologías conceptualmente industriales.

La creciente demanda de videojuegos hace que cada vez más a menudo se creen productos con falta de coherencia y/o cohesión, sobre todo en el sector indie-comercial, los cuales ofrecen conglomerados de middlewares (modelos 3D genéricos en este caso) sin ningún sentido. Es importante que el usuario tenga videojuegos indie coherentes con identidad y carácter a precios asequibles que compitan con los proyectos indie-comerciales.

La idea es que posea un entorno único, de estética realista estilo cyberpunk, rico en detalles, con calidad de luces y texturas, con identidad propia que no use ningún modelo 3D genérico o comercial. Este se ejecutará en Windows y se jugará con mando de Xbox.

1.3. Objetivos generales

Listado y descripción de los objetivos del TF, ordenados por relevancia.

1.3.1. Objetivos principales

Objetivos de la aplicación/producto/servicio:

- Crear un videojuego dotado con menú de navegación y un nivel jugable
- Crear un videojuego con mecánicas de *ARPG* y vista isométrica
- Ejecutable con SOs Windows 10 en adelante.
- Compatible con mando de XBOX

Objetivos para el cliente/usuario:

- Conseguir un aspecto espectacular de los entornos y personajes
- Crear efectos visuales atractivos de interacción (armas, elementos del GUI)

Objetivos personales del autor del TF:

- Importarlos modelos de producción propia a Unity con éxito (escala y rigging)
- Animar personajes y entornos de producción propia en *Unity*
- Texturizar modelos 3D en *Substance 3D Painter*
- Modelar personajes y entornos en 3D ricos en detalle
- Bocetar personajes, objetos y entornos

1.3.2. Objetivos secundarios

Objetivos adicionales que enriquecen el TF:

- Usar la imaginación para crear un entorno y personajes originales
- Hacer un uso no abusivo de los *assets* para crear el proyecto
- Combinar diferentes competencias de campos distintos para conseguir un objetivo específico (diseño, programación, sonido, ilustración, modelado, texturizado, competencias comunicativas, uso de las TIC, *storytelling*)
- Combinar diferentes softwares para la creación del proyecto (*Blender, Unity, Substance 3D Painter, Photoshop, Illustrator, Visual Studio*)

1.4. Metodología y proceso de trabajo

En este apartado se define cual es la metodología de gestión de proyectos escogida para llevar a cabo el desarrollarlo, así como los diferentes hitos a alcanzar durante todo este desde cero hasta la entrega del mismo, su memoria y presentación.

La metodología es una ágil, ya que según un estudio de 2017 del PMI (*Project Manager Institute*), las metodologías ágiles ya suponían un 71% de las metodologías de desarrollo elegidas por las organizaciones de diseño de software con una tendencia positiva en auge.

Concretamente se ha escogido la SCRUM. La razón es porque es una metodología de carácter incremental (como el desarrollo de software que va sumando funciones) y organizativo, ya que divide sus hitos en *sprints* (tareas cortas en bloques cortos de tiempo).

Por lo tanto, los procesos o hitos que se describirán a continuación serán encapsulados en *sprints*. Todos salvo la memoria que es un hito que se realizará de manera transversal durante todo el proyecto (menos su revisión final, antes de la entrega que será un hito más).

Lista de procesos:

- Análisis del encargo
- Ideación del proyecto
- Crear un borrador inicial
- Crear repositorio en GitHub
- Estudio de mercado
- Formarse para modelar en *Blender*, crear los Riggs en *Blender* y animar en *Blender*
- Formarse para texturizar en *Substance 3D Painter*
- Formarse sobre el “*character controller*” y el “*animator*” de Unity (*Unity Learn*)
- Crear un *moodboard* inspiracional
- Definir una paleta de colores
- Definir una paleta tipográfica, tamaños y estilos de fuentes
- Ilustrar bocetos de los entornos (suelos, paredes, ...)
- Ilustrar los bocetos de los objetos (armas, herramientas, comunes, ...)
- Ilustrar los bocetos de los personajes
- Creación de *wireframes*
- Búsqueda de *plugins* para *Blender*
- Búsqueda de *assets* de *Unity*

- Modelar los entornos
- Modelar los objetos
- Modelar los personajes
- Texturizar los entornos
- Texturizar los objetos
- Texturizar los personajes
- Crear el proyecto de *Unity*, configurarlo y exportar todos los contenidos (propios y de terceros)
- Montar, programar y diseñar una versión alfa del juego incluyendo flujo básico pantallas
- Diseñar elementos del *GUI* en *Photoshop* e *Illustrator*
- Implementar diseños en los elementos del GUI
- Mejorar ya aumentar el número de enemigos del nivel
- Crear el diseño de interacción visual y retroacciones del GUI
- Implementar las mejoras en los efectos (*VFX* y *VFX*)
- Testeo y optimización de una primera beta
- Revisión de la memoria
- Testeo y optimización de una versión definitiva
- Entrega de proyecto y memoria
- Realización video presentación
- Defensa virtual ante el tribunal

Por otra parte, para terminar, aunque ya se han ido definiendo a través de la memoria, se detallan a continuación en las siguientes tablas los recursos, métodos y técnicas que se utilizarán para desarrollar el proyecto.

Recursos:

<i>Software, assets, plugins</i>	Uso
<i>Assets</i>	Proporcionar una base sobre la que programar el juego
<i>Blender</i>	Programa para modelar, animar y texturizar en 3D
<i>Illustrator</i>	Programa para crear elementos del GUI en 2D
<i>Photoshop</i>	Programa para diseñar y componer elementos del GUI en 2D
<i>Substance 3D Painter</i>	Programa usado para texturizar los modelos
<i>Unity</i>	Motor grafico usado para montar y

Prototipo de videojuego para PC compatible con mando de Xbox, clase *ARPG*, estilo *Cyberpunk*, con vistas isométricas

	desarrollar el videojuego
<i>Visual Studio</i>	Editar <i>scripts</i>

Tabla 1: Recursos (*software, assets, plugins*)

Metodologías o técnicas:

Método o técnica	Uso
Animación 3D	Para animar a los personajes y objetos
Diseño gráfico	Para crear imágenes en 2D y texturas
Diseño de interfaces	Para crear el <i>GUI</i>
Estudio de mercado	Para definir con más concreción que se necesita donde
Ilustración clásica	Para crear los bocetos
Modelado 3D	Para crear escenarios, objetos y entorno
<i>Moodboard</i>	Para inspirarse
SCRUM	Método ágil para gestionar el desarrollo del proyecto

Tabla 2: Métodos o técnicas

1.5. Planificación

En este apartado se especifica como se divide o reparten cada uno de los procesos en Sprints, su duración, así como las fechas de inicio y comienzo de cada una de estas. Todas englobadas en hitos parciales en función a las entregas de las PECs.

PEC 1 - Plan de Proyecto			
Hitos / Milestones / Sprints	Duración	Día inicio	Día Final
Análisis del encargo	6 días	mie 28.09.22	lun 03.10.22
Ideación del proyecto	3 días	mar 04.10.22	jue 06.10.22
Crear el borrador inicial	3 días	vie 07.10.22	dom 09.10.22

Tabla 3: Planificación PEC1

PEC 2 – Estudio de mercado y primera versión del proyecto			
Hitos / Milestones / Sprints	Duración	Día inicio	Día Final
Crear repositorio en GitHub	0 días	lun 10.10.22	lun 10.10.22
Estudio de mercado	5 días	lun 10.10.22	vie 14.00.00
Definir target objetivo (<i>buyer</i> personas)	4 días	sab 15.10.22	mar 18.10.22
Formarse para modelar, crear los Riggs y animar en <i>Blender</i>	1 días	mie 19.10.22	mie 19.10.22
Formarse para texturizar en <i>Substance 3D Painter</i>	1 días	jue 20.10.22	jue 20.10.22
Formarse sobre el “ <i>character controller</i> ” y el “ <i>animator</i> ” de Unity (<i>Unity Learn</i>)	1 días	vie 21.10.22	vie 21.10.22
Crear un <i>moodboard</i> inspiracional	1 días	sab 22.10.22	sab 22.10.22
Definir una paleta de colores	1 días	dom 23.10.22	dom 23.10.22
Definir una paleta tipográfica, tamaños y estilos de fuentes	1 días	lun 24.11.22	lun 24.11.22
Ilustrar bocetos de los entornos, objetos y personajes.	3 días	mar 25.11.22	jue 27.11.22
Creación de <i>wireframes</i>	1 días	vie 28.11.22	vie 28.11.22
Búsqueda de <i>plugins</i> para <i>Blender</i> y assets para Unity	1 días	sab 29.11.22	sab 29.11.22
Modelado 1: Modelar parte de los	1 días	dom 30.11.22	dom 30.11.22

entornos, objetos y personajes			
Texturizado 1: Texturizar parte de los entornos, objetos y personajes	1 días	lun 31.11.22	lun 31.11.22
Crear el proyecto de <i>Unity</i> , configurarlo y exportar todos los contenidos (propios y de terceros)	1 días	mie 01.12.22	mie 01.12.22
Montar, programar y diseñar una versión alfa del juego incluyendo flujo básico pantallas	13 días	mie 02.12.22	lun 14.11.22

Tabla 4: Planificación PEC2

PEC 3 - Implementación de versión jugable			
Hitos / Milestones / Sprints	Duración	Día inicio	Día Final
Modelado 2: Modelar el resto de los entornos, objetos y personajes	4 días	mar 15.11.22	vie 18.11.22
Texturizado 2: Texturizar el resto de los entornos, objetos y personajes	4 días	sab 19.11.22	mar 22.11.22
Diseñar elementos del <i>GUI</i> en <i>Photoshop</i> e <i>Illustrator</i>	2 días	Mie 23.11.22	jue 24.11.22
Implementar diseños en los elementos del <i>GUI</i>	2 días	vie 25.11.22	sab 26.11.22
Mejorar y aumentar el número de enemigos del nivel	6 días	dom 27.11.22	vie 02.12.22
Crear el diseño de interacción visual y retroacciones del <i>GUI</i>	6 días	sab 03.12.22	jue 08.12.22
Implementar las mejoras en los efectos (<i>VFX</i> y <i>VFX</i>)	6 días	vie 09.12.22	mie 14.12.22
Testeo y optimización de una primera beta	4 días	jue 15.12.22	dom 18.12.22
Entrega	0 días	dom 18.12.22	dom 18.12.22

Tabla 5: Planificación PEC3

PEC 4 - Memoria y productos finales			
Hitos / Milestones / Sprints	Duración	Día inicio	Día Final
Testeo y optimización de una versión definitiva del prototipo	18 días	lun 19.12.22	jue 05.01.23
Realización video presentación	3 días	vie 06.01.23	dom 08.01.23
Redacción y revisión de la memoria	5 días	lun 09.01.23	sab 15.01.23

Entrega de proyecto, memoria y video presentación	0 días	dom 15.00.23	dom 15.01.23
---	--------	--------------	--------------

Tabla 6: Planificación PEC4

PEC 5 - Defensa Virtual - Tribunal de TF			
<i>Hitos / Milestones / Sprints</i>	Duración	Día inicio	Día Final
Defensa virtual ante el tribunal	7 días	mie 25.01.23	mie 01.02.23

Tabla 7: Planificación PEC5

1.5.1. Diagrama de gantt

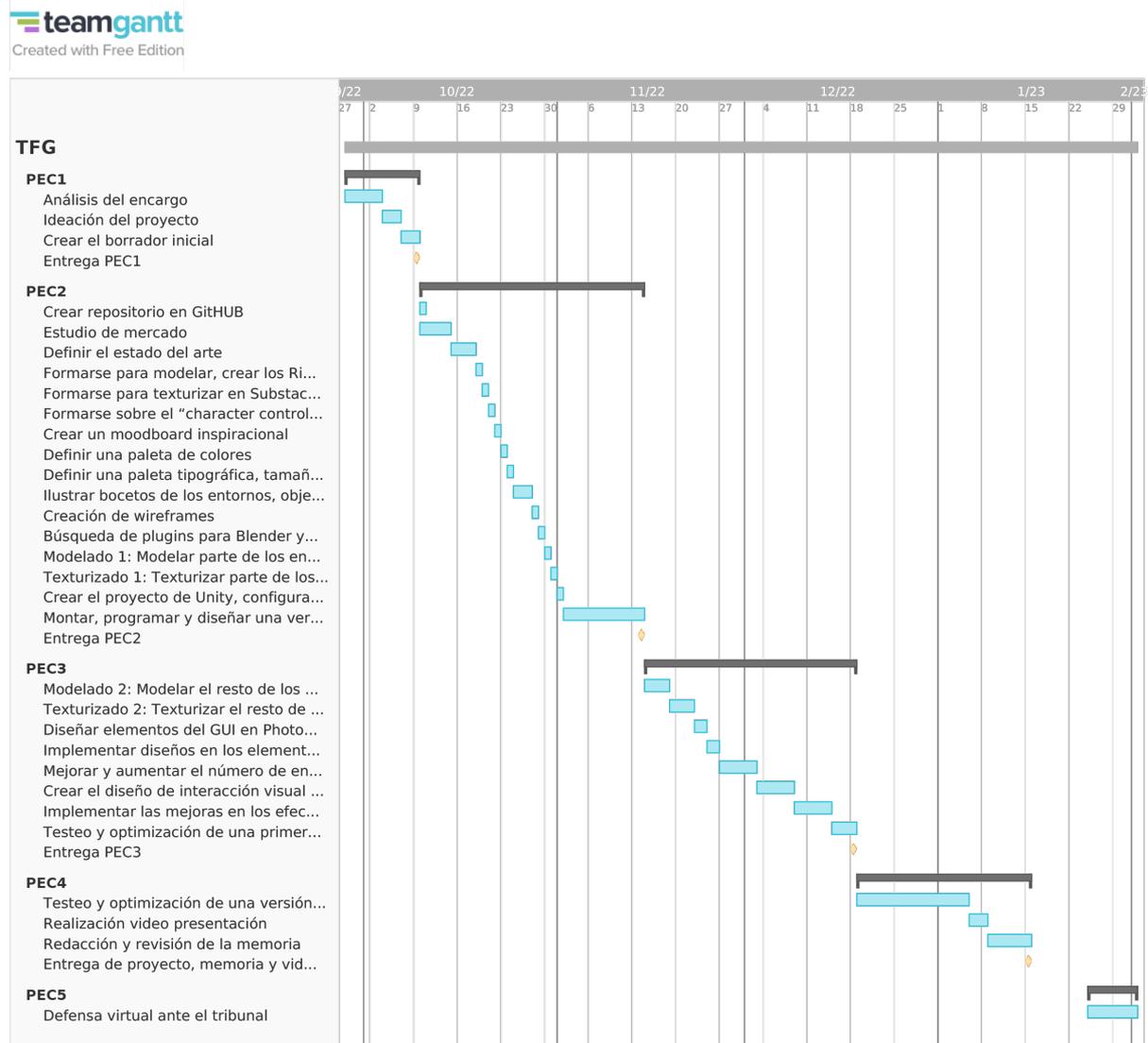


Figura 5: Diagrama de gantt

1.6. Presupuesto

Para la realización del proyecto se necesitan contemplar dos tipos de gastos principales:

- Los gastos derivados de la adquisición de programas y herramientas (activos o bienes)
- Los gastos derivados de las tareas (recursos humanos)

No es necesario contemplar los gastos derivados de adquirir el equipamiento técnico ya que forman parte del patrimonio inicial del proyecto.

<i>Software, assets, plugins</i>	Coste (estimativo)
<i>Assets</i>	50 €
<i>Plugins Blender</i>	30€
<i>Blender</i>	Gratis
<i>Adobe creative cloud suit</i>	(61,95 €/mes) x 3 meses
<ul style="list-style-type: none"> • <i>Illustrator</i> • <i>Photoshop</i> • <i>Substance 3D Painter</i> 	185,85 €
<i>Unity</i>	Gratis
<i>Visual Studio</i>	Gratis
TOTAL	265 €

Tabla 8: Costos derivados de la adquisición de *software, assets y plugins*

Tarrea	Tiempo	Coste
<i>Análisis y estudio de mercado</i>	32	960 €
<i>Definición de target (buyer personas)</i>	32	960 €
<i>Animación</i>	32	960 €
<i>Diseño de interfaz</i>	32	960 €
<i>Diseño de entornos, objetos y personajes</i>	32	960 €
<i>Modelado</i>	40	1.200 €
<i>Texturizado</i>	40	1.200 €
<i>Programación</i>	120	3600 €
TOTAL		16.800 €
*Precio por hora único a 30€		

Tabla 9: Costes derivados de los recursos humanos

1.7. Estructura del resto del documento

A modo de referencia, en este apartado se incluye una pequeña guía de contenidos para el lector donde se explica que contenido contiene cada capítulo. Todo con la intención de que se tenga una visión más global del conjunto de procesos, y de las relaciones que estos tienen entre sí.

En el **apartado dos** de del documento (**análisis de mercado**), se hace un estudio para determinar cuál es la situación actual y en que contexto se encuentra el producto. También nos sirve para definir un público objetivo y evaluar las fortalezas y debilidades de la competencia (DAFO).

En el **apartado tres** (**propuesta**), se define con más precisión cuáles serán las especificaciones y objetivos del producto. Cuál será el modelo de negocio que adaptará el título, así como cuál será su estrategia de marketing.

En el **apartado cuatro** (**diseño**) se tratan los aspectos relevantes a la arquitectura de la información, así como la estructura general de la aplicación o servicio. Pero también se tratará el tema de la interfaz, su estilo y usabilidad, al igual que aspectos que conciernen al diseño de la programación.

En el **quinto apartado** (**implementación**) se dan las instrucciones necesarias para poder ejecutar el juego y la guía para su correcta ejecución

En el **sexto** (**demostración**) se dan las instrucciones detalladas de uso, se hace un recorrido por los distintos prototipos que se realizaron y se describen los test que se llevaron a cabo junto con su metodología.

En el apartado **séptimo** (**conclusiones y líneas de futuro**) se exponen las síntesis finales derivadas de todo el proceso de realización del TFG y se traza un camino por el cual el producto seguirá desarrollándose y evolucionando tras la entrega del prototipo inicial finalizado.

Para terminar, tenemos el apartado de **bibliografía** y los **anexos** estándar correspondientes a la calidad de esta memoria.

2. Análisis de mercado

En este capítulo se hará un análisis de la situación actual del mercado. Para ello, empezaremos comentando el estado actual del mercado en torno a los juegos de PC, luego se define la audiencia potencial y por último se analiza a la competencia.

2.1. Estado del mercado

En la actualidad y desde hace tiempo, el mercado de la compra y venta de videojuegos para ordenador se realiza mayoritariamente desde internet. Cada vez es menos común que los usuarios hagan compras en soporte físico ya que a diferencia de la digital, ocupa lugar, pueden romperse y no se pueden clonar. Además, hoy en día existen páginas webs que se dedican a la reventa de códigos de juegos de diversas plataformas, entre las cuales se pueden destacar: *Steam (Valve)*, *Origin (EA)*, *Battel.net (Blizzard)* y *Epic Games*.

Si bien como hemos visto existen cantidad de plataformas de venta y descarga de juegos en la nube, hay una en concreto que destaca sobre las demás tanto en antigüedad como en innovación. Esta es la plataforma de *Steam*, nacida originalmente en 2003 como un programa que proveía actualizaciones automáticas de sus juegos, aunque posteriormente acabaron incluyendo juegos de terceros. Hoy en día tiene el mayor catálogo de todas las plataformas, ofrece antipiratería, sistemas anti-trampas, *streamings*, servidores de emparejamiento, servicios de redes sociales y últimamente incluso han creado su propio ordenador de juegos con su propio sistema operativo incluido, la *SteamDeck* con el *SteamOS*.

Una vez concretada la plataforma objetivo de distribución y el porqué, vamos a hablar de la situación actual del catálogo de videojuegos para *Steam*. Por el momento, si realizamos una búsqueda en este, podremos encontrar gran cantidad de títulos de *RPG* (rol estático, de mouse y teclado) con vistas isométricas algunos con temática *cyberpunk* (*Gamedec*, *Shodowrun Retuns*, *Beautiful Desolation...*) y otros no. Pero vemos que donde realmente existe una carencia de títulos en el catálogo es del género **ARPG** (rol de acción) **isométricos**, concretamente ***Twin-Stick Shooters***, con temática ***cyberpunk***.

Los *Twin-Stick Shooters*, son los que utilizan los dos *joysticks* o palancas de control del mando para controlar al personaje (izquierdo/movimiento, derecho/apuntar). Actualmente existen diversos géneros clásicos que se mezclan con estos para crear distintos **conjuntos de géneros** que dan como resultados productos muy interesantes, de los cuales se pueden destacar principalmente las siguientes combinaciones:

- ***Twin-Stick Shooters* + arcade + vista isométrica** (ej. *Meltdown*)

- **Twin-Stick Shooters + supervivencia + crafting + vista isométrica** (ej. *The Last Stand: Aftermath*)
- **Twin-Stick Shooters + ARPG + vista isométrica** (ej. “*The Ascent*”)

Pero donde no se ven tantos títulos es en el último “conjunto de géneros” nombrado, donde al parecer **existe una oportunidad para llenar el vacío** con títulos de calidad que se alejen de las “estéticas collage” que nombrábamos en apartados anteriores.

Por otro lado, comentar la tendencia de algunos de títulos del estilo *cyberpunk* en ofrecer acabados visuales muy impactantes como componente básica de la combinación del éxito de estos. Ésto es en buena parte posible, gracias al aumento del rendimiento y eficiencia así del hardware como del software contemporáneo, los cuales ofrecen incrementos exponenciales de la calidad gráfica por cada nueva generación o versión. Con la suma del nivel de la competencia que eso conlleva.

2.2. Público objetivo y perfiles de usuario (*Buyer persona*)

Para continuar y antes de analizar a la competencia, se delimitará el público objetivo a nivel general según las características distintivas más relevantes para el proyecto:

- **Medios técnicos / económicos:** destinado a usuarios con un poder adquisitivo medio y ordenadores *PC master race* (*Ben Croshaw, 2008*) o de gama media/alta capaces de ejecutar videojuegos AAA o que exijan de un alto rendimiento y que además dispongan de un mando de Xbox o copia autorizada para controlar el juego.
- **Edad/Madurez:** destinado (generalmente) a usuarios adultos, especialmente de 25 años de edad en adelante o a aquellos con la capacidad de asimilar la complejidad de la trama y los conceptos filosóficos que se plantearían en una futura versión del juego que no se contempla en este primer prototipo jugable.
- **Culturales contextuales:** los contenidos culturales incluidos en el juego estarán principalmente centrados en un público occidental, pese a que el juego en si tenga ciertas influencias orientales.
- **Temática/Estilo:** dirigido a usuarios apasionados del género *cyberpunk* y la ciencia ficción, con grandes dosis de entornos industriales, *cyborgs* y robots basados en las inspirados por relatos como los que dejó el escritor *William Gibson* en sus libros; o películas como “*Akira*” (*Katsuhiro Otomo, 1988*), “*Ghost in the Shell*” (*Masamune Shirow, 1989*) o “*Blade Runner*” (*Ridley Scott, 1982*).

A continuación, y teniendo en cuenta las características que definen al usuario objetivo en general, se facilitan una serie de perfiles ficticios o *buyer* personas basados en ellos que nos permitan contextualizar mejor a que tipo de usuario va dirigido el juego.

2.2.1. Buyer personas

<h1>David</h1> <p>Repartidor (800 €/mes) Adulto joven (25 años)</p>		
<p>Breve bibliografía:</p> <p>David vive en casa de su madre. Prácticamente invierte todo su salario en videojuegos y dispositivos o periféricos de juegos. Le encanta todo lo que está relacionado con la ciencia ficción y la tecnología. Siempre anda en la busca de nuevos videojuegos de su estilo preferido, cyberpunk.</p>		
<p>Datos demográficos:</p> <p>Ubicación: Madrid Nivel de educación: ESO Situación sentimental: soltero</p>	<p>Intereses:</p> <p>Tecnología Cyberpunk Música electrónica</p>	
<p>Hábitos:</p> <p>Jugar videojuegos Ver series y películas Escuchar música</p>	<p>Objetivos:</p> <p>Crear un gaming room épico Conseguir el mayor numero de juegos Probar videojuegos nuevos</p>	

Tabla 10: Buyer person 1 - David

<h1>Emily</h1> <p>Programadora (3000 USD/mes) Adulto (28 años)</p>		
Breve bibliografía: <p>Emily pasa 8 horas al día en la empresa. Cuando termina de trabajar y para desconectar le gusta jugar ARPG de accion de vista isométrica. Adora ver a los personajes corriendo en su pantalla gigante a toda resolución desde esa perspectiva. Siente que casi los puede tocar y fluye jugando mientras vacía su mente de códigos.</p>		
Datos demográficos: <p>Ubicación: New York Nivel de educación: Universitarios Situación sentimental: Soltera</p>	Intereses: <p>Ciencia Programación Innovación</p>	
Hábitos: <p>Ver películas y series Jugar videojuegos Salir con amigas</p>	Objetivos: <p>Tener un trabajo entretenido Ganar un buen sueldo Divertirse en su tiempo libre</p>	

Tabla 11: Buyer person 2 - Emily

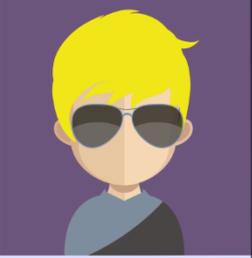
<h1>Ulrich</h1> <p>Ingeniero mecánico (4000 €/mes) Adulto joven (30 años)</p>		
Breve bibliografía: <p>Ulrich vive solo en un estudio de ciudad. Teletrabaja 8 horas al día, a veces 10 y el resto del tiempo lo dedica a jugar videojuegos y a configurar su casa con domótica. Posee un muy potente ordenador destinado a gráficos y diseño.</p>		
Datos demográficos: <p>Ubicación: Berlín Nivel de educación: Universitarios Situación sentimental: soltero</p>	Intereses: <p>Mecánica Videojuegos Domótica</p>	
Hábitos: <p>Programar Jugar videojuegos Construir artefactos caseros</p>	Objetivos: <p>Tener un trabajo interesante Ganar bien Gastar el tiempo libre en videojuegos</p>	

Tabla 12: Buyer person 3 - Ulrich

<h1>Buffy</h1> <p>Entrenadora def. personal (3000 €/mes) Adulta (36 años)</p>		
Breve bibliografía: <p>Buffy es una chica de acción. Cuando no esta entrenando, esta dando clases y cuando no esta jugando videojuegos. Se podría decir de ella que es una adicta a la adrenalina y que le encanta estar siempre activa. Le gusta manejar armas tanto en la vida real como en los videojuegos. Su película favorita es Terminator 2.</p>		
Datos demográficos: <p>Ubicación: Londres Nivel de educación: Universitarios Situación sentimental: Pareja</p>	Intereses: <p>Artes marciales Armas Acción</p>	
Hábitos: <p>Entrenar Jugar videojuegos Estar en actividad constante</p>	Objetivos: <p>Vivir el momento Vivir aventuras Ser una gran maestra marcial</p>	

Tabla 13: Buyer person 4 - Buffy

<h1>Joe</h1> <p>Vj, Dj y productor (4000 USD/mes) Adulto (36 años)</p>		
Breve bibliografía: <p>Joe vive con su novia en un apartamento. Pasa casi todo el día mezclando, produciendo música electrónica o clips de video para sus sesiones. Le apasiona la tecnología y en casa tiene un estudio propio con gran cantidad de equipamiento profesional. En sus ratos libres le encanta jugar juegos con vista isométrica de estilo cyberpunk.</p>		
Datos demográficos: <p>Ubicación: Georgia Nivel de educación: universitarios Situación sentimental: pareja</p>	Intereses: <p>Tecnología (especialmente multimedia) Corriente Cyberpunk Arte</p>	
Hábitos: <p>Producir material audiovisual Jugar videojuegos isométricos Pasar tiempo en pareja</p>	Objetivos: <p>Disfrutar la vida al máximo Hacer cosas con pasión Adquirir tecnología, música y juegos</p>	

Tabla 14: Buyer person 5 - Joe

<h1>Gabriella</h1> <p>Profesora idiomas (2000 €/mes) Adulta (33 años)</p>		
Breve bibliografía: <p>Gabriella se define a si misma como una Freaky. Colecciona comics japoneses.(idioma que habla), muñecos de sus heroes favoritos y videojuegos. Le encanta sobre todo el genero Cyberpunk, del cual apenas deja escapar algun título sin comprar.</p>		
Datos demográficos: <p>Ubicación: Roma Nivel de educación: Universitarios Situación sentimental: Pareja</p>	Intereses: <p>Coleccionismo Cyberpunk Animes</p>	
Hábitos: <p>Comprar en internet Jugar videojuegos Hacer tandems</p>	Objetivos: <p>Conocer culturas nuevas Viajar Recrearse en mundos fantásticos</p>	

Tabla 15: Buyer person 6 - Gabriella

<h1>Pierre</h1> <p>Proyeccionista de cine (3000 €/mes) Adulto (32 años)</p>		
Breve bibliografía: <p>Pierre vive con su novia y su gato en un apartamento de París. Está fascinado con el género Cyberpunk y ve en él cierta corriente filosófica que le resulta muy interesante. También disfruta recesándose en los ambientes industriales complejos llenos de detalles, ya que sacian su mente, la cual siempre ansia procesar información compleja.</p>		
Datos demográficos: <p>Ubicación: París Nivel de educación: CFGS Situación sentimental: pareja</p>	Intereses: <p>Cyberpunk Filosofía Cultura</p>	
Hábitos: <p>Leer Jugar videojuegos Ver películas y series.</p>	Objetivos: <p>Estudiar el género cyberpunk Crear su propio subgénero Escribir sus propias historias</p>	

Tabla 16: Buyer person 7 - Pierre

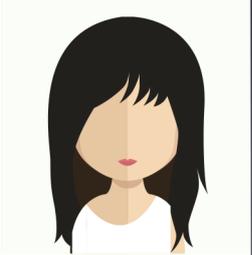
<h1>Lucía</h1> <p>Ingeniera robótica (3500 €/mes) Adulta joven (25 años)</p>		
Breve bibliografía: <p>Lucía es una gran apasionada de la ciencia y de la ciencia y ficción. Acaba de acabar la carrera y ha encontrado un empleo para diseñar robots. Como también es una gran apasionada de los videojuegos, a menudo le gusta jugar títulos de ciencia ficción que le puedan servir de inspiración para sus proyectos de la vida real.</p>		
Datos demográficos: <p>Ubicación: Barcelona Nivel de educación: Universitarios Situación sentimental: Soltera</p>	Intereses: <p>Tecnología Ciencia Videojuegos</p>	
Hábitos: <p>Leer Jugar videojuegos Ver películas y series</p>	Objetivos: <p>Posicionarse y ascender en su trabajo Ser una gran creativa Crear un robot humanoide</p>	

Tabla 17: Buyer person 8 - Lucía

2.3. Análisis de la competencia

Para continuar con nuestro análisis de mercado, haremos un análisis de la competencia basado en 5 de los principales competidores (directos e indirectos) que se encuentran en la plataforma de Steam. Todos ellos comparten la característica de ser ***Twin-Stick Shooters + ARPG + vista isométrica***.

Estos son:

- ***The Ascent***
- ***Ruiner***
- ***I see red***
- ***Synthetik 2***
- ***Jydge***

2.3.1. The Ascent



Figura 6: *The Ascent*. Nominado al juego "Outstanding Visual Style" en el Steam Awards del año 2021. Fuente: Steam

Breve descripción del producto y su contexto social, económico y empresarial:

En 2021 "Neon Giant" lanzó "*The Ascent*", el cual es el primer título lanzado por la nueva empresa indie desarrolladora de videojuegos con sede en Suecia. Pese a que la empresa es nueva, 2 de los fundadores principales de esta son veteranos de la industria de los juegos AAA (Tor Frick y Arcade Berg), habiendo trabajado en varios títulos de renombre anteriormente. Por lo que la empresa cuenta con una fuerte base de conocimientos y experiencia en el sector.

Después del lanzamiento de su juego recibieron un gran aluvión de buenos comentarios y algunos malos en respuesta a éste, ya que el juego aun poseía ciertos bugs, algo comprensible sabiendo que es su primer proyecto. Todos ellos fueron compensados por la gran calidad visual de los entornos, animaciones y modelados, así como con las actualizaciones posteriores las cuales fueron corrigiendo todos los errores. Aun así, solo en su primera semana de ventas recaudaron 5 millones de dólares.

Comparativa de las características:

- **Precio producto:** 29.99 €
- **Reseñas generales en Steam:** Mayormente positivas 71 % (13.290 reseñas)
- **Lanzamiento:** 2021
- **Servicios ofrecidos:**

- Compatible con mando
- Modo historia un jugador
- Modo historia dos jugadores online y pantalla compartida
- **Edad usuario objetivo:** +18
- **Estilo y diseño:** el juego es impresionante visualmente. Rico en detalles. Es tan bueno que fue nominado al juego "Outstanding Visual Style" de 2021. Motor Unreal Engine 4.
- **Usabilidad/Jugabilidad:**
 - Cantidad de armas y habilidades configurables.
 - Control fluido y dinámico
- **La calidad del producto:** es un producto de una calidad excelente, visualmente sublime, muy optimizado y que supero los límites gráficos del momento implementando sus propios algoritmos dentro del motor gráfico *Unreal Engine 4*.
- **Modelo de pago:** compra del juego inicial más DLC.
- **Marketing:** video tráiler y uso de la comunidad, redes sociales, creadores de contenido, *streamers* e *influencers* para dar a conocer el juego.

2.3.2. *Ruiner*



Figura 7: *Ruiner* (2017). Fuente: Steam

Breve descripción del producto y su contexto social, económico y empresarial:

Ruiner (lanzado en 2017) es el primer título de la empresa independiente Reikon Games ubicada en Polonia (fundada en 2014). El equipo de la empresa ya trabajo en títulos de renombre con anterioridad como *The Witcher 1 a 3*, *Dead Island* o *Dying Light* y algunos de sus colaboradores en otros como *This War of Mine*, *Anomaly*, *Shadow Warrior*, *Sniper*, *Hard Reset* o *Cyberpunk 2077*. Cuentan con un gran equipo de talentos tanto nuevos como debutantes.

Comparativa de las características:

- **Precio producto:** 19.99 €
- **Lanzamiento:** 2017
- **Reseñas generales en Steam:** Muy positivas 90 % (8.382 reseñas)
- **Servicios ofrecidos:**
 - Compatible con mando
 - Modo historia un jugador
- **Edad usuario objetivo:** +18

- **Estilo y diseño:** el juego cuenta con una estética industrial llena de detalles. Algunos de los materiales tienen un acabado entre metal y plástico muy interesante. El motor utilizado fue el *Unreal Engine 4*.
- **Usabilidad/Jugabilidad:**
 - Cantidad de armas y habilidades configurables. Gran flexibilidad de configuraciones y estilos de juego.
 - Control fluido, pero algo extraño ya que el personaje no gira el ángulo de visión con el movimiento del personaje, se ha de hacer de manera extra con el joystick derecho.
- **La calidad del producto:** muy buena en general.
- **Modelo de pago:** compra del juego inicial más DLC.
- **Marketing:** video tráiler y uso de la comunidad, redes sociales, creadores de contenido, *streamers* e *influencers* para dar a conocer el juego.

2.3.3. *I see red*



Figura 8: *I see red* (2022). Fuente: Steam

Breve descripción del producto y su contexto social, económico y empresarial:

I see red es el título más fresco de todos, lanzado en octubre de este mismo año (2022) por la empresa indie “*Whiteboard Games*” establecida en Buenos Aires, Argentina. Este es su primer trabajo y no cuentan con personal con experiencia en títulos AAA ni famosos del sector, pero si con el patrocinio de la aceleradora de proyectos Neutrón.

El trabajo realizado por este pequeño equipo para ser su primer producto es destacable. Nació en origen como un proyecto universitario que tras el grado y dos años después, acabó convirtiéndose en una realidad.

Comparativa de las características:

- **Precio producto:** 9.99 €
- **Lanzamiento:** 2022
- **Reseñas generales en Steam:** positivas 90 % (32 reseñas)
- **Servicios ofrecidos:**
 - Compatible con mando
 - Modo historia un jugador

- **Edad usuario objetivo:** A diferencia de los otros títulos, este se centra en un público más adolescente de a partir de 16 años y que podría llegar hasta los 30 años.
- **Estilo y diseño:** Su una estética un poco más estilo *Fornite* en blanco y negro con grandes acentos de rojo que dirigen la atención de la acción del juego. El motor utilizado es desconocido.
- **Usabilidad/Jugabilidad:**
 - Interacción con el entorno.
 - Control fluido
- **Calidad del producto:** buena calidad tanto gráfica como técnica en general a falta de cierta optimización.
- **Modelo de pago:** compra del juego inicial más DLC.
- **Marketing:** video tráiler y uso de la comunidad.

2.3.4. Synthetik 2



Figura 9: Synthetik 2 (2021). Fuente: Steam

Breve descripción del producto y su contexto social, económico y empresarial:

Synthetik 2 (2021) es el segundo título de la empresa alemana situada en Berlín *Flow Fire Games*. Con anterioridad publicaron *Synthetik: Legion Rising* y su expansión multijugador *Synthetik Arena*. La empresa actualmente se dedica a desarrollar su tercer título. Es a su vez desarrolladora y distribuidora de sus productos y ha recibido distintos premios, entre ellos de Steam y de la gigante asiática Tencent.

Comparativa de las características:

- **Precio producto:** 19.99 €
- **Lanzamiento:** 2017
- **Reseñas generales en Steam:** Muy positivas 81 % (950 reseñas)
- **Servicios ofrecidos:**
 - Compatible parcialmente con mando
 - Modo historia un jugador
- **Edad usuario objetivo:** +12
- **Estilo y diseño:** el juego cuenta con una estética industrial simplista que juega mucho con las paletas de colores opuestos y complementarios. El motor que utiliza es Unity.

- **Usabilidad/Jugabilidad:**
 - Cantidad de tecnologías y arsenales.
 - Control pulido y táctico.
- **La calidad del producto:** Presenta un acabado brillante y es de muy buena calidad.
- **Modelo de pago:** compra del juego única.
- **Marketing:** video tráiler y uso de la comunidad, redes sociales, etc.

2.3.5. Jydge



Figura 10: *Jydge* (2017). Fuente: Steam

Breve descripción del producto y su contexto social, económico y empresarial:

Jydge fue lanzado en 2017 por la ya veterana empresa 10tons Ltd. La cual tiene residencia en Tampere, Finlandia. Esta empresa fue fundada en 2003 y desde entonces ha sacado decenas de títulos al mercado (22 en *Steam*). Muchos de ellos son **Twin-stick shooter+vista isométrica** por lo que tienen una fuerte base y amplia experiencia en el desarrollo de este tipo de géneros.

Comparativa de las características:

- **Precio producto:** 14.95 €
- **Reseñas generales en Steam:** Mayormente positivas 88 % (372 reseñas)
- **Lanzamiento:** 2017
- **Servicios ofrecidos:**
 - Compatible con mando
 - Modo historia un jugador
 - Modo historia dos jugadores online y pantalla compartida
 - Jugador contra jugar (PVP)
- **Edad usuario objetivo:** +12

- **Estilo y diseño:** pese a que la temática es *cyberpunk* su estilo es un poco más *low poly* y menos rica en detalles que la del resto de competidores. Pese a ello luce un aspecto fabuloso y representa perfectamente la esencia del género.
- **Usabilidad/Jugabilidad:**
 - Cantidad de armas y habilidades configurables.
 - Control fluido en entornos destruibles.
- **La calidad del producto:** Es un producto bien definido que ofrece gráficos espectaculares con modelos *low poly* para ordenadores no tan potentes.
- **Modelo de pago:** compra única del juego.
- **Marketing:** uso de la comunidad, redes sociales, creadores de contenido, *streamers* e *influencers* para dar a conocer el juego.

2.4. Análisis DAFO

Es este subapartado se detallan tanto las características internas del proyecto (Debilidades y Fortalezas) , tanto como las de su situación externa (Amenazas y Oportunidades).

2.4.1. The Ascent

	De origen interno	De origen externo
Negativos	<ul style="list-style-type: none">- Juego corto.- Storytelling algo flojo.	<ul style="list-style-type: none">- Ha ganado muchos competidores
Positivos	<ul style="list-style-type: none">- Innovación gráfica.- Uso de grandes recursos tecnológicos.- Rejugable.	<ul style="list-style-type: none">- Juego debut de la empresa, oportunidad de crear una primera buena impresión.- Gran recibimiento de la comunidad.- Es un referente gráfico desde que salió.

Tabla 18: DAFO The Ascent

2.4.2. Ruiner

	De origen interno	De origen externo
Negativos	<ul style="list-style-type: none"> - Juego corto. - Música algo molesta. - Control del personaje algo raro. 	<ul style="list-style-type: none"> - Salió hace años y tiene muchos nuevos competidores.
Positivos	<ul style="list-style-type: none"> - Estética y efectos increíbles. - Uso de cantidad de recursos tecnológicos. - Rejugable 	<ul style="list-style-type: none"> - Juego debut de la empresa, oportunidad de crear una primera buena impresión - Buen recibimiento de la comunidad. - Buena crítica en Steam.

Tabla 19: DAFO Ruiner

2.4.3. I see red

	De origen interno	De origen externo
Negativos	<ul style="list-style-type: none"> - Estética un tanto arriesgada/inusual (blanco y negro con rojo). 	<ul style="list-style-type: none"> - Tiene grandes competidores actualmente en el mercado. - Ha tenido poco alcance.
Positivos	<ul style="list-style-type: none"> - Entorno altamente interactivo. - Gran de variedad de habilidades. - Precio competente. 	<ul style="list-style-type: none"> - El producto ha entrado en dos plataformas de venta de juegos muy conocidas (<i>Steam</i> y <i>Epic Games</i>).

Tabla 20: DAFO I see red

2.4.4. Synthetik 2

	De origen interno	De origen externo
Negativos	<ul style="list-style-type: none"> - Falta de optimización. - Compatibilidad parcial con mando. 	<ul style="list-style-type: none"> - Ha tenido poco alcance.
Positivos	<ul style="list-style-type: none"> - El producto tiene una estética muy llamativa. - La estética favorece las mecánicas del juego y hace que sean más entendibles creando una cohesión muy interesante. - Gran variedad de configuraciones de juego. - Rejugable. 	<ul style="list-style-type: none"> - Tiene muy buena crítica en Steam. - Muy buena acogida de la comunidad.

Tabla 21: DAFO Synthetik 2

2.4.5. Jydge

	De origen interno	De origen externo
Negativos	<ul style="list-style-type: none"> - Falta de optimización. - Compatibilidad parcial con mando. 	<ul style="list-style-type: none"> - Ha tenido poco alcance. - Es un poco antiguo y tiene nuevos y más innovadores competidores en el mercado.
Positivos	<ul style="list-style-type: none"> - El juego está muy bien optimizado. - Se ve muy bien para los pocos recursos que consume. - Entorno altamente interactivos. 	<ul style="list-style-type: none"> - Tiene muy buena crítica en Steam. - Muy buena acogida de la comunidad.

Tabla 22: DAFO Jydge

2.5. Mapa conceptual (*competitive landscape*)



Figura 11: Mapa conceptual (*competitive landscape*)

3. Propuesta

En base al análisis de mercado, se procede a explicar de manera resumida la propuesta. Se hará especial hincapié en las aquellas particularidades que lo diferencian de la competencia.

3.1. Definición de objetivos/especificaciones del producto

A continuación, se detallan los objetivos principales del videojuego.

Objetivos fundamentales:

- Ofrecer una estética novedosa como sello de identidad del producto dentro del subgénero de ciencia-ficción, *cyberpunk*.
- Crear unas mecánicas de juego fluidas y de calidad similar a la de los juegos de la competencia
- Ofrecer un precio competitivo y distintivo dentro del segmento del sector (6,99 €)

Listado detallado de las características:

- Menú de navegación y un nivel jugable
- Ejecutable en SOs Windows 10 en adelante.
- Modo un jugador
- Compatible con mando de XBOX
- *Twin-Stick Shooters + ARPG + vista isométrica*
- Uso de armas
- Entorno destruible
- Entorno dinámico (movimiento de ciertas partes)
- Esquema de iluminación estilo neón con luces duras

3.2. Modelo de negocio

En la actualidad existen cantidad de modelos de negocios para los videojuegos. Algunos de ellos regalan el juego y luego piden dinero por servicios, otros ni siquiera se pueden descargar a menos que hagas una compra digital. En este subapartado, pasaremos a describir los principales modelos de negocio y sus principales características para tener una macro visión del panorama actual y de la razón del modelo de negocio.

- **Advergaming:** son aquellos juegos que se crearon con el objetivo de promocionar un producto, como por ejemplo los juegos de automovilismo con marcas reales como el *Need for Speed* de la distribuidora EA, o el antiguo *Pepsiman*.
- **Try before you buy + Compra digital :** son aquellos que te permiten descargar una demo jugable del juego con funciones limitadas para captar al usuario.
- **Compra digital:** Un clásico. Una única compra online que te permite acceder a todo el contenido del juego.
- **Compra digital + DLC:** uno de los modelos clásicos por excelencia. Se requiere de la compra digital del juego para poder jugarlo. El usuario tiene la opción de ampliar su contenido comprando digitalmente DLCs (*Downloadable content*). Estos se suman al contenido ya existente para expandirlo.
- **Free to play/Freemium:** dentro de este modelo existen varios métodos por los cuales el empresario puede obtener ingresos del usuario:
 - Micropagos: por bienes dentro del juego (skins, armas, créditos del juego,...).
 - Suscripciones: donde regalan bienes, créditos dentro o ventajas dentro del juego.
 - Pases de temporada: como pases premium que regalan bienes o créditos dentro del juego.
- **Pay to play:** este es menos común en occidente, pero es uno de los modelos más populares en Corea del Sur, donde se venden tarjetas de crédito con tiempo de juego o vidas del personaje. Es la evolución de los salones recreativos convencionales. Destinado a usuarios casuales.
- **Play-to-Earny NFT:** este modelo de negocio se basa en otorgar al usuario activos digitales en el juego, los cuales pueden ser incrementados de valor en función del tiempo que invierten los jugadores y que luego se pueden vender por dinero real. Han ganado más relevancia después de las necesidades sufridas durante la pandemia.

De todas las descritas, se ha elegido el modelo de **Compra única** dado que a nivel profesional es el primer juego de ordenador que publicaría en Steam y es el modelo de negocio por excelencia, descendiente de las antiguas formas de venta de soportes físicos en las tiendas de juegos en los tiempos donde las TIC aun no estaban tan a la vanguardia de la sociedad.

Se ha pensado un precio de **6,99 €** que seria capaz de poner el producto en el top número uno en precio dentro de la competencia.

3.3. Estrategia de marketing

La estrategia de marketing es exactamente la misma que se viene usando en el mundillo *gamer*. El juego hará uso de la comunidad de *Steam*, de su búsqueda con filtros o etiquetas, de las reseñas, de

Prototipo de videojuego para PC compatible con mando de *Xbox*, clase *ARPG*, estilo *Cyberpunk*, con vistas isométricas

los creadores de contenido que realicen videos de *gameplays* del juego y los compartan por las diversas redes sociales y en general de la red para darse a conocer. No se pretende invertir dinero alguno en publicidad.

4. Diseño

En este capítulo, se tratan todos los aspectos relevantes a la conceptualización y desarrollo del prototipo del juego. Estos abarcan todos los procesos productivos de diseño llevados a cabo, desde lo más técnico, llegando hasta lo más artístico y creativo.

4.1. Arquitectura general del juego

La arquitectura que compone un videojuego, contemplado desde un punto de vista técnico teórico simplista, no es más que la combinación de un conjunto de elementos o *assets* (diversos tipos de contenidos multimedia) manejados y coordinados por un software, o más concretamente, motor gráfico. Para representar este concepto y a la vez adecuarnos al entorno de trabajo, tomaremos el modelo conceptual de Unity, el cual a grosso modo cataloga en 6 grandes grupos los *assets* principales que maneja.

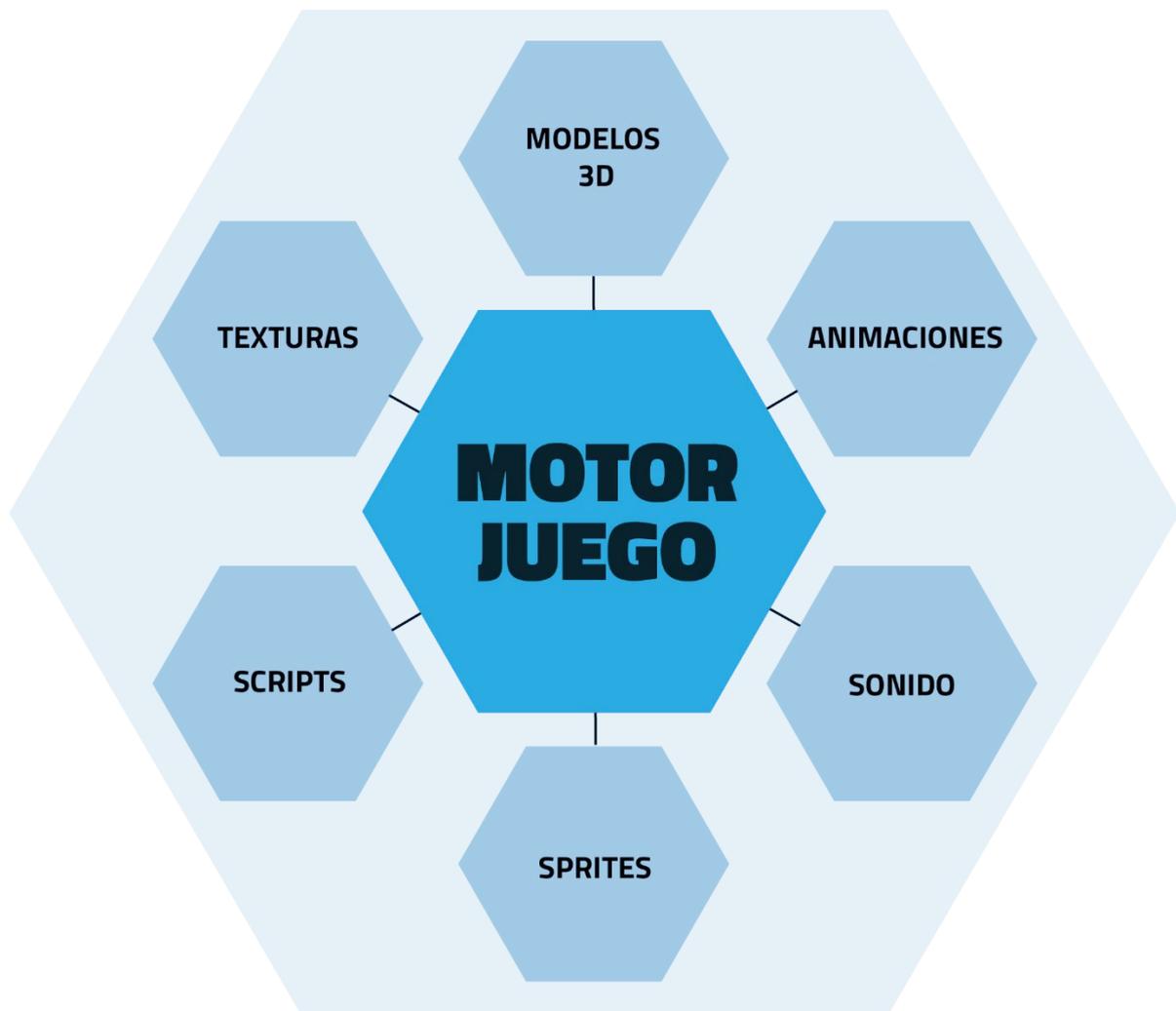


Figura 12: Partes principales de un videojuego (según Unity Learn)

4.2. Arquitectura detallada del videojuego y sus funciones

A continuación, se hará una descripción detallada reforzada con diagramas de los elementos principales que componen la aplicación.

4.2.1. Arquitectura interna del juego

El videojuego a nivel interno está compuesto por escenarios, objetos de juego y de los componentes que pueden ir asociados a él, todos ellos organizados por una jerarquía. A continuación, se describen las características y usos de cada uno dentro del programa.

Para comenzar y yendo en orden descendente, la primera gran división del juego son sus escenas o **scenes**. Estas son los elementos más básicos en los que se divide el programa de juego. Permiten crear y gestionar los diferentes apartados principales de este. Pueden ser tales como la pantalla de carga, el propio escenario de juego, el menú inicial o el de carga de niveles.

A su vez, estas contienen **game objects**, Estos son objetos dentro del juego y pueden ser de diferentes tipos como pudieran ser la cámara principal, un trozo de suelo, el botón de un canvas, un personaje, una imagen, una luz, etc. En resumen, cualquier objeto que es observable desde la jerarquía (último párrafo de este subapartado).

Los **games objects**, así mismo albergan en su interior los **components** (componentes) necesarios para ser gestionados. Éstos, les otorgan los distintos comportamientos y funcionalidades a los objetos. Por nombrar algunos de los más comunes podemos encontrar el *collider* (usado para hacer que un objeto choque con otro) y el *rigid body* (que convierte un objeto en sólido, con masa), aunque existen muchos más.

Paralelamente a los **games objects** se encuentran los **prefabs**. Estos son objetos del juego que ya vienen configurados con sus propios componentes. Suelen venir con los **assets** y pueden ser tales como un personaje que ya lleva incorporado los componentes que lo animan y hacen tener masa, rigidez, así como la capacidad de saltar o sprintar.

Para terminar, la **hierarchy** o jerarquía les da prioridad, ámbito y relación a todos los elementos. Esto es una característica común en todos los lenguajes de programación y tan solo se trata de cómo están ordenados y anidados unos objetos dentro de otros. Un ejemplo cotidiano podría ser de nuevo el personaje, que además porta un arma. Por lo tanto, a donde vaya el personaje ira el arma, ya que es controlado por el objeto padre que anida a ambos.



Figura 13: Diagrama de los componentes del juego, relación y jerarquía

En la figura 13 se pueden observar distintos tipos de ejemplos de cómo pueden estar organizados y relacionados todos los elementos de las formas más comunes. Para la realización de este videojuego, en concreto se han hecho uso de los siguientes elementos, entre los cuales encontramos escenas y *game objects* con sus componentes tanto de Unity como del *asset/motor "Top Down Engine"*:

- **Scene:** contiene todo en su interior, desde los scripts hasta los componentes o modelos usados para crear las pantallas de carga, así como los menús jugables.

- **Game Manager:** *script* administrador responsable de la tasa de fotogramas, del bucle del juego, de guardar puntos en la escena y maneja la escala de tiempo. Opcionalmente puede llevar la cuenta de las vidas del personaje.
- **Level Manager:** *script* que genera al jugador, gestiona los puntos de control y reaparición.
- **Input Manager:** *script* que gestiona las acciones realizadas con el mando.
- **Event System:** componente que se encarga de enviar eventos a objetos en la aplicación en función de la entrada, en nuestro caso, un mando.
- **Standalone Input Module:** este componente funciona como se esperaría que funcionara una entrada de mando o ratón. Eventos para presionar botones y similares se envían en respuesta a la entrada.
- **GUIManager:** *script* que gestiona los diferentes splash del juego: HUD, pause y game over entre otros.
- **Inventory:** *script* que gestiona el contenido del inventario, añadiendo, borrando o equipando ítems.
- **Inventory Input Manager:** *script* que gestiona las acciones realizadas con el mando mientras está el menú del inventario abierto.
- **Camera:** componente que funciona como cámara principal del juego que captura el mundo recreado en la escena y lo muestra al jugador.
- **Cinemachine:** componente de cámara inteligente para efectos creativos, incluye entre otros:
 - **Cinemachine Brain:** componente vínculo entre la cámara de Unity y las cámaras virtuales de *cinemachine* en la escena. Es donde se definen las reglas para la fusión de cámaras virtuales.
 - **CinemachineVirtualCamera:** componente que crea una cámara virtual.
 - **Cinemachine Confiner:** *script* que limita la posición de la cámara a un volumen o área determinado.
 - **Cinemachine Impulse Listener:** *script* que hace que la cámara seleccionada reaccione a las señales de impulso (*impulse sources*) y reaccione sacudiendo la cámara.
- **MM Cinemachine Camara Shaker:** *script* que añade movimiento orgánico a la cámara usando los métodos de Top Down Engine para manipular Cinemachine.
- **MMCinemachineZoom:** *script* que permite aplicar zooms usando los métodos de Top Down Engine para manipular Cinemachine.
- **Flare Layer:** componente que añade destellos a la cámara.
- **Light:** componente que añade luces a la escena.
- **Audio Listener:** componente que recibe el audio de una fuente de la escena y la reproduce por los altavoces.

- **Post-process Volume:** componente que añade filtros estáticos que generan sensación de volumen como el de oclusión de ambiente, efecto *bloom*, viñeta, profundidad de campo, aberraciones cromáticas, distorsiones de lente y gradientes de color.
- **MMVignetteShaker:** *script* que se activa bajo determinadas circunstancias y añade temporalmente el efecto de agitar el marco oscuro de viñeta del juego (con los disparos o acciones).
- **MMLensDistorsionShaker:** *script* que se activa bajo determinadas circunstancias y añade temporalmente el efecto de agitar la cámara (explosiones, disparos, etc..).
- **MMDepthOfFieldShaker:** *script* que se activa bajo determinadas circunstancias y añade temporalmente el efecto de variar la profundidad de campo del juego.
- **MMColorGradingShaker:** *script* que se activa bajo determinadas circunstancias y añade temporalmente un gradiente de color radial sobre la pantalla (al morir).
- **MMChromaticAberrationShaker:** *script* que se activa bajo determinadas circunstancias y añade temporalmente el efecto de aberración cromática sobre la pantalla (disparos explosiones).
- **MMBloomShaker:** *script* que se activa con los disparos y agita el marco oscuro de viñeta del juego (disparos explosiones).
- **MMAutoFocus:** *script* que enfoca automáticamente al personaje o carácter jugable.
- **Post-Process Layer:** componente que añade filtros de postproceso de imagen como *Anti-aliasing* (FXAA), etc..
- **NavMeshSurface:** *script* que hace coincidir al agente o jugador con las superficies del juego. Decide si este debe de saltar o si una zona es transitable o no.
- **Mesh Renderer:** componente que permite renderizar una maya deformable.
- **Mesh Collider:** componente que permite detectar colisiones en base a una maya de referencia.
- **Sprite Renderer:** componente que renderiza un *sprite* y determina como se muestra visualmente.
- **Rect Transform:** componente para representar un elemento es su interior y usar en la interfaz.
- **Canvas:** componente representa un espacio abstracto en el cual se renderiza el GUI.
- **Canvas Group:** componente que permite controlar ciertos aspectos de un grupo de elementos que conforman una GUI como la transparencia o la interactividad.
- **Canvas Scaler:** componente que permite escalar todo el canvas respetando sus proporciones.
- **Canvas Renderer:** componente que permite renderizar un *game object* con un canvas.
- **Vertical Layout Group:** componente que permite organizar elementos de forma vertical en un GUI.
- **Button:** componente que añade un botón a un GUI.
- **Image:** componente que añade una imagen a un GUI.

- **PauseButton:** script que determina el comportamiento de la pulsación sobre el botón pause.
- **BackgroundMusic:** script que permite seleccionar una canción de fondo.
- **MM Sound Manager:** script que hace las veces de mezclador de canales de audio. Gestiona el volumen de los efectos, de la música, de la interfaz y del volumen maestro.
- **MM Time Manager:** script que recibe los MMtimeScaleEvents y MMFreezeFrameEvents alterando el tiempo del juego.
- **Character Controller:** script que permite realizar movimientos con el personaje restringidos por colisiones.
- **Rigidbody:** script que otorga la propiedad de ser un cuerpo rígido a un objeto del juego.
- **TopDown Controller 3D:** script que controla el *character controller* y lo adapta a los métodos de "Top Down Engine".
- **Character:** script donde se implementan todas las reglas del juego de tu personaje, como las siguientes:
 - **Character Grid Movement:** script que permite moverse al personaje sobre un malla.
 - **Character Orientation 3D:** script que permite orientar al personaje seleccionando el tipo de *input*.
 - **Character Movement:** script que permite el movimiento del personaje.
 - **Character Crouch:** script que permite que el personaje se ponga de rodillas.
 - **Character Run 3D:** script que permite que nuestro personaje corra.
 - **Character Jump 3D:** script que permita que nuestro personaje salte.
 - **Character Inventory:** script que permite que el personaje acceda al inventario.
 - **Character Handle Weapon:** script que permite que el personaje maneje un arma.
 - **Character Pause:** script que permite al personaje entra en el modo de *pause*.
 - **Character Time Control:** script que permite variar el tiempo.
 - **Character Rotate Camera:** script que permite rotar la cámara sobre el personaje.
- **Health:** script que permite que un objeto tenga vida.
- **Animator:** componente que permite animar a un objeto del juego como podría ser un personaje.
- **Character Animation Feedbacks:** scripts que dispara *feedbacks* en el animador como el sonido o las partículas.
- **AIBrain:** script que determina el comportamiento de las inteligencias artificiales del juego.
- **MMProgressBar:** script para generar barras de carga como la vida o el estado de carga de una pantalla.
- **MMFollowTarget:** script que permite a un objeto del juego perseguir a otro.
- **Item Picker:** script que permite que un objeto que sea recogido y que determina que ítem se entrega al inventario del jugador.

Mencionar también los *prefabs* (game objects + componentes pre armados) más destacables usados para la realización de este proyecto.

- **3DCameras:** *prefab* con la configuración cámaras básicas.
- **Loft Suit:** *prefab* con la configuración básica de un personaje jugable (player 1).
- **PatrolAndMoveAI:** *prefab* con la configuración de un enemigo que patrulla y se mueve.
- **LoftPatrolAndShootAI:** *prefab* con la configuración de un enemigo que patrulla y dispara.
- **PatrolSeekAndDestroyAI:** *prefab* con la configuración de un enemigo que patrulla, se detiene y dispara.
- **PatrolSeekAndDestroyReloadingAI:** *prefab* con la configuración de un enemigo que patrulla, se detiene, dispara y recarga
- **Stimpack:** *prefab* con la configuración de un elemento que al cogerlo que da vida al jugador.
- **WeaponCrateHandgun:** *prefab* con la configuración de un objeto que al recogerlo da una pistola.
- **WeaponCrateShotgun:** *prefab* con la configuración de un objeto que al recogerlo da una escopeta.
- **WeaponCrateAssaultRifle:** *prefab* con la configuración de un objeto que al recogerlo da un rifle de asalto.
- **WeaponCrateSniperRifle:** *prefab* con la configuración de un objeto que al recogerlo da un rifle de asalto.
- **AmmoCrateHandgun:** *prefab* con la configuración de un objeto que al recogerlo da munición de pistola.
- **AmmoCrateShotgun:** *prefab* con la configuración de un objeto que al recogerlo da munición de escopeta.
- **AmmoCrateAssaultRifle:** *prefab* con la configuración de un objeto que al recogerlo da munición del rifle de asalto.
- **AmmoCrateSniperRifle:** *prefab* con la configuración de un objeto que al recogerlo da una munición del rifle *sniper*.
- **AiBrain:** script que determina el comportamiento de las IA.
- **AIDecisionDetectTargetRadius3D:** *script que determina el radio de detección de los enemigos.*
- **AIActionMovePatrol3D:** script que permite a las IAs patrullar.
- **AIActionShoot3D:** script que permite disparar a las IAs en entornos 3D.
- **AIActionDoNothing:** script que permite a el personaje no hacer nada y estar quieto de pie.
- **AIActionPathfinderToTarget3d:** script que permite a las IAs ir detrás del personaje jugable trazando una ruta hacia su posición.

4.2.2. Navegación

En este subapartado se aporta un diagrama ilustrativo de navegación que representa todos los caminos posibles entre las distintas pantallas y menús del juego.

La figura 14 muestra como al abrir el juego, lo primero que hace éste es mostrar la pantalla de carga para luego dejarnos en la pantalla de *start*. Aquí se puede hacer uso del ratón para abrir los *settings* o presionar el botón A para continuar.

Después de mostrar otra pantalla de carga distinta a la inicial, acabamos en la escena de seleccionar nivel. Con la cruceta del mando nos posicionamos sobre el nivel que queremos jugar y presionando el botón A encima de éste entraremos en él.

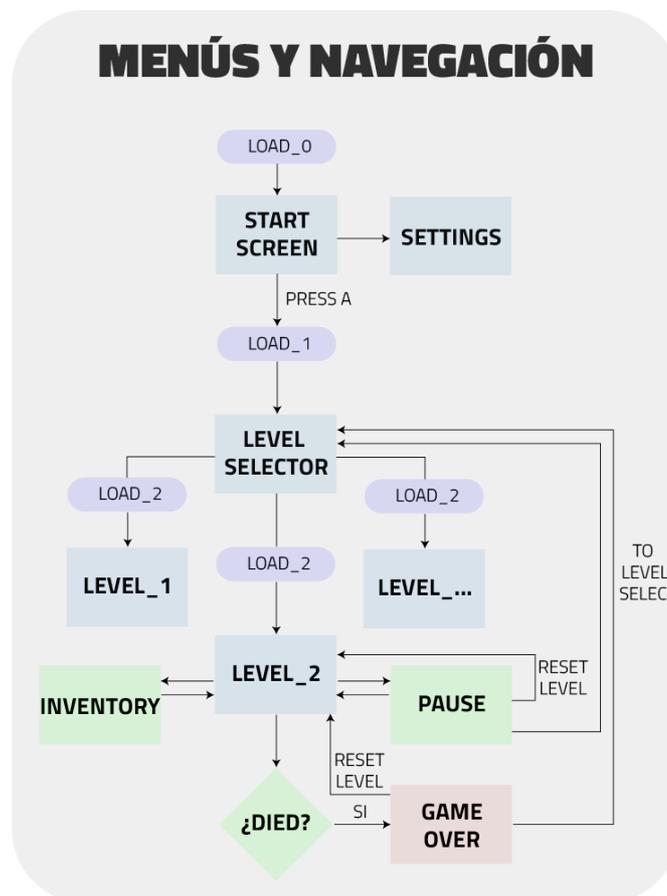


Figura 14: Menús y navegación del juego

Una vez dentro de un nivel, tenemos la posibilidad de abrir el inventario para gestionar los ítems o de abrir el menú de pause. Abriendo el menú de pause tenemos la opción de reiniciar el nivel o ir al selector de niveles. De igual manera se accede a una pantalla de *game over* (cuando el personaje

muere) donde también tenemos las mismas opciones que en el de pause, es decir, resetear el nivel o cambiarlo.

4.2.3. Contenidos

En cuanto a los contenidos que forman parte del programa, tenemos los 6 elementos que maneja el motor Unity (figura 12) para ejecutar un juego. Detallamos a continuación el software que se usará y formato de cada uno de ellos.

- **Modelos 3D:** software Blender o Maya (.fbx)
- **Textura:** software Substance Painter (.mat, .sbsar, .mdl)
- **Animaciones:** Blender (.anim)
- **Sonido:** Adobe Audition (.ogg, .flac, .mp3)
- **Sprites:** Illustrator, Photoshop (.png)
- **Script:** Visual Studio (.cs)

4.3. Diseño gráfico, de interfaces y de jugabilidad

En este apartado se aportan distintos tipos de contenidos que han servido para diseñar tanto el apartado artístico y visual, como el que concierne a la experiencia de juego y jugabilidad del prototipo.

4.3.1. Estilos

Para definir la estética y carácter gráfico o visual del producto se han hecho uso de los siguientes recursos.

- *Moodboard*
- Bocetos y esbozos
- Logotipos y anagramas
- Paleta de colores
- Paleta tipográfica, tamaño y estilo de fuentes
- Fuentes, iconos, botones y otros elementos gráficos

Para mayor comodidad y claridad todos ellos vienen en un documento aparte, la primera versión básica del “Manual de identidad” creado para el producto y anexado en esta memoria.

4.3.2. Jugabilidad (*Twin-stick shooter*)

El juego se ha diseñado siguiendo los estándares de usabilidad y los conceptos anclados de los clásicos *Twin-stick shooter*. Es decir, un personaje que es controlado con las 2 palancas analógicas de un mando (una dirección de movimiento y otra de apuntado del arma). A continuación, se comparte una infografía que detalla los controles del juego.

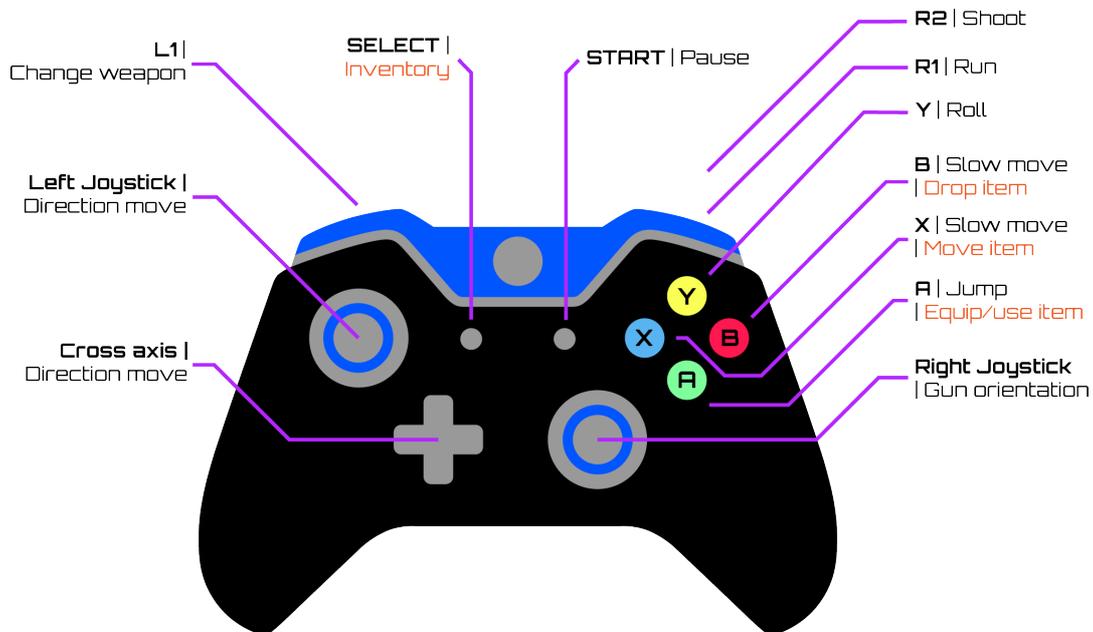


Figura 15: Infografía controles del prototipo

A nivel de accesibilidad y ya que el juego está pensado para ser jugado exclusivamente con mando de Xbox, se han tenido en cuenta los botones más usados para configurar los controles tanto del personaje, como de la navegación, así como de operatividad en el inventario. Las acciones que el jugador podrá realizar pulsando los botones del inventario, están descritas principalmente con un texto que además indica el botón a usar. Ya que muchos jugarán con el mando original, o con una copia que respete los colores de los botones, estos están además identificados con colores asociativos en los botones en pantalla para pulsar con el ratón. Se puede apreciar como en la figura 16, los tres botones en pantalla (*Equip/use*, *Move* y *Drop*) corresponden en acciones y color con los del mando de la figura 15.

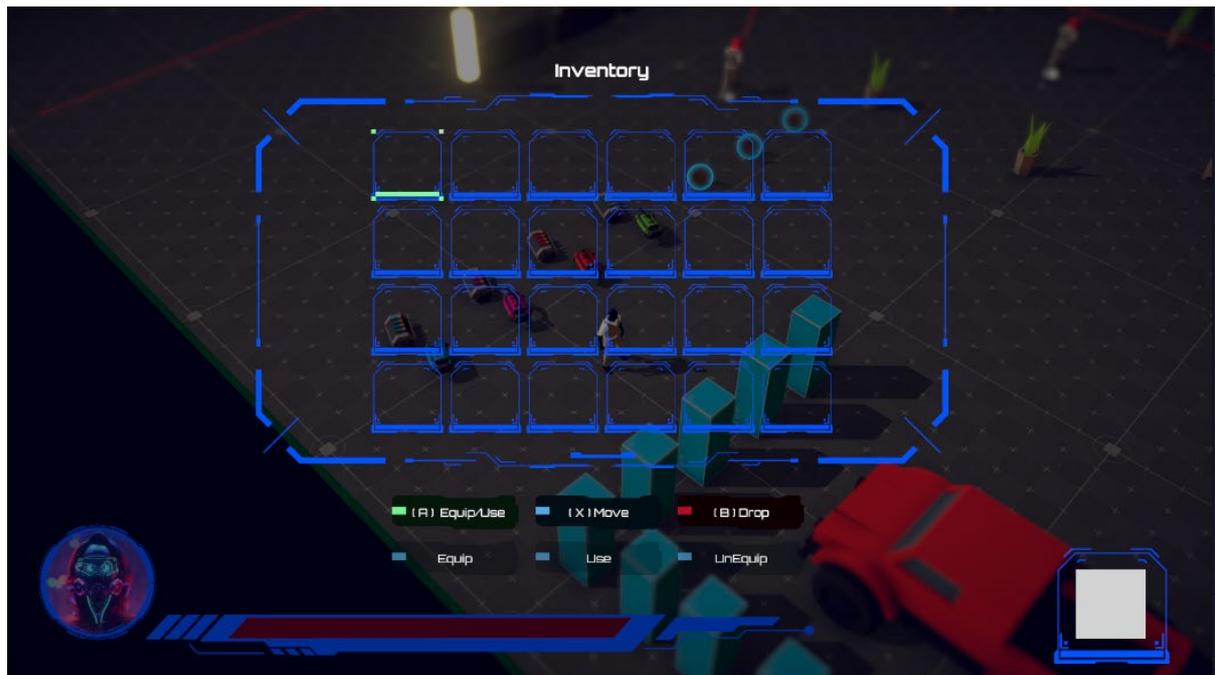


Figura 16: Inventario junto con los botones con etiquetas identificativas y colores.

Agregar que, pese a que también es posible usar teclado y ratón, el juego no se puede jugar bien sin mando, ya que se necesita de dos joysticks para poder apuntar con precisión y llevar a cabo las mecánicas de juego fundamental del género. De nuevo se insiste en comentar que el prototipo está pensado para ser compatible con el mando de Xbox.

4.4. Motor gráfico y assets. DDCs e IDEs

Para la realización del prototipo se ha hecho uso de un motor gráfico, un solo *asset* de tercerizados y varios *DDCs* (*Digital Content Creators*) o creadores de contenido digitales, así como un IDE (*Integrated Development Environment*).

De la gran variedad de motores gráficos que existen en la actualidad, se ha escogido Unity para el proyecto por su relativa sencillez de metodología de trabajo e intuitivo interfaz. Además, existen cantidad de *assets* de tercerizados que nos pueden ahorrar mucho tiempo, sobre todo en cuanto a scripts se refiere. Los scripts en Unity se escriben en C# (*CSharp*) y son fundamentales para determinar los comportamientos de los escenarios y de los objetos dentro del juego (*Game objects*).

Para la realización del prototipo se hizo uso del *asset* “*Top Down Engine*” el cual ofrece cantidad de scripts y prefabs que han ahorrado mucho tiempo en el desarrollo del proyecto. Tales como pantallas de carga, inventarios, gestores de entradas del mando entre otros tantos componentes indispensables para el buen y correcto funcionamiento de un juego que cumpla las expectativas mínimas de calidad.

Por otro lado, y de manera transversal tenemos los programas de creación de contenido o *DDCs*. Estos me han permitido tanto editar los contenidos de tercerizados como crear nuevos completamente desde cero.

Por último, tenemos la *IDEs*, que son los entornos integrados pensados para el desarrollo de software o creación y modificación de *scripts*.

En concreto se han usado los siguientes *DDCs* y el siguiente *IDE* por las siguientes razones :

<i>Software, assets, plugins</i>		Uso
<ul style="list-style-type: none"> • <i>Blender y Maya</i> 	3D DDC	Programas predilectos por excelencia usados por la comunidad de creadores de contenido para modelar y animar en 3D.
<ul style="list-style-type: none"> • <i>Illustrator</i> • <i>Photoshop</i> • <i>Substance 3D Painter</i> 	2D DDC	<p><i>Illustrator</i> y <i>Photoshop</i> son programas estándar del sector para crear elementos del GUI en 2D incluido en el plan de <i>creative cloud</i>.</p> <p><i>Substance</i> es desde hace mucho un gran programa de texturizado que ha sido recientemente adquirido por Adobe y está incluido en el <i>creative cloud</i>.</p>
<ul style="list-style-type: none"> • <i>Audition</i> 	Audio DDC	Programa incluido en el plan de <i>creative cloud</i> .
<ul style="list-style-type: none"> • <i>Visual Studio Code</i> 	<i>IDEs</i>	Programa gratuito y muy optimizado de Microsoft para la edición de <i>scripts</i> .

Tabla 23: *DDCs* e *IDEs* usados en el proyecto y la razón

En cuanto a hardware para el desarrollo del prototipo, he de comentar que el juego se ha desarrollado con un PC con las siguientes características:

- OS WIN 10
- 16 GB de RAM

- CPU AMD Ryzen 9 3900X 12-Core Processor (3.80 GHz)
- GPU AMD Radeon RX 5700 (8GB)

4.5. Diseño del nivel

En este subapartado se detalla toda la información que concierne al diseño de nivel. Estos son:

- Cómo se ha diseñado el nivel y con que meta.
- Que elementos son de terceros y cuales son propios.
- Que elementos interactivos posee.

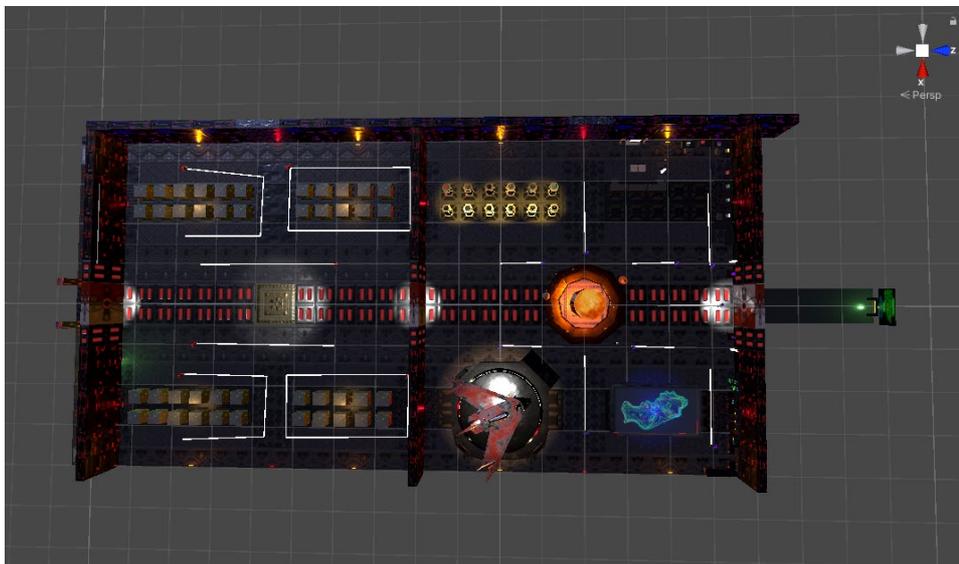


Figura 17: Nivel visto desde arriba en Unity

4.5.1. Idea original

Para diseñar el nivel, primero se pensó en que tipo de reto se podría enfrentar el jugador para esta primera versión del prototipo, que no fuera demasiado largo, pero que fuera totalmente funcional y a la vez muy vistoso. También era importante que fuera divertido y que la jugabilidad fuera buena.

Para ello primero se concedió la idea de dividir el nivel en 4 segmentos separados por grandes portones:

1. Primera zona/zona inicial: pequeña zona donde el jugador comienza en una zona tranquila que le permita empezar a familiarizarse con los controles, el entorno y los primeros objetos que encuentra. También hay un enemigo un tanto despistado que servirá para familiarizar al jugador con el combate y las coberturas (dos pilares con vida).

2. Segunda Zona: Llena de elementos destruibles y unos cuantos enemigos que permitan al jugador empezar a soltarse con el control. La idea también es que empiece a experimentar los AFX y VFX incluidos en el juego a la vez que compruebe el comportamiento del inventario (ya que al llenarse no recogerá más objetos).
3. Tercera zona: diseñada para mostrar la estética del juego con más solidez que las anteriores a la vez que incluye enemigos de mayor rango e incluso un jefe que vigila la puerta final. Es el zenit tanto de la parte visual como de acción y pretende llamar la atención de los posibles interesados.
4. Cuarta zona/zona final: para terminar, se incorporó un pequeño pasillo con un mensaje al final que indica al jugador que el juego se ha acabado por el momento y que vendrán más actualizaciones.

4.5.2. Primeros bocetos

En este subapartado, se exponen los tres bocetos realizados para el proyecto. Estos están en negativo para resaltar los contornos y mejorar la visualización.

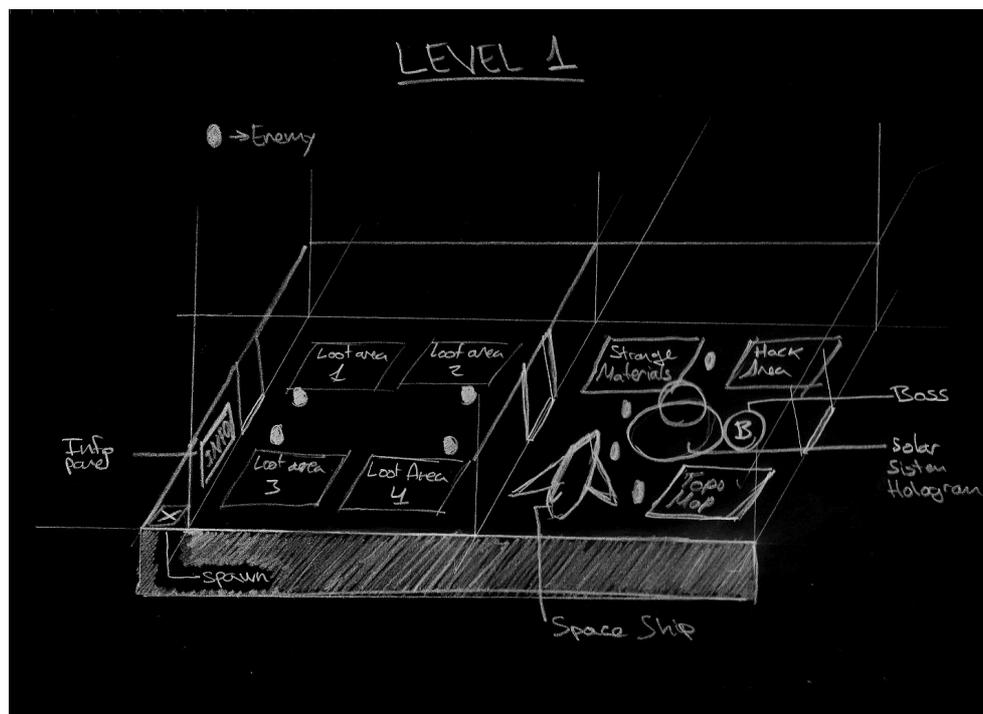


Figura 18: Boceto del Nivel 1

En la Figura 16 se contempla el primer concepto de nivel que considera las proporciones y dimensiones, divisiones, las áreas temáticas, además de los enemigos. Si bien las disposiciones y número de enemigos no quedaron exactamente igual, sirvieron de punto de partida para empezar a diseñar el nivel con una visión inicial.

A continuación, dos bocetos más que muestran el pilar expositor de materiales extraños y la mesa de proyección holográfica para la visualización de mapas topográficos o diagramas estilo ciencia ficción. Comentar que hay algunas creaciones, como la nave espacial y su stand, que no tienen bocetos originales, si no que se diseñaron sobre la marcha.

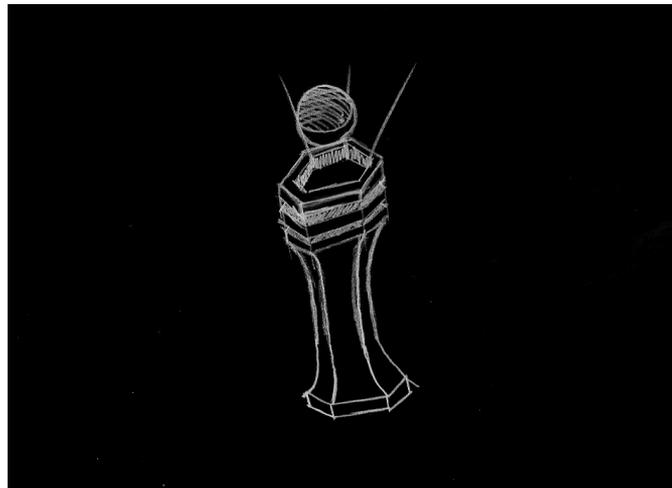


Figura 19: Boceto de pilar de materiales

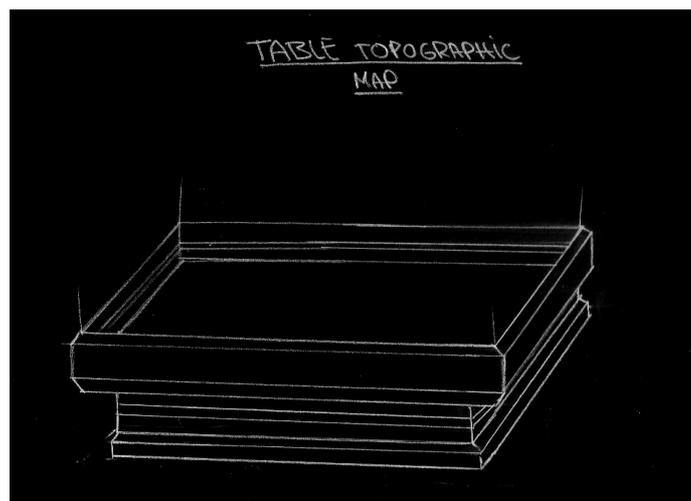


Figura 20: Boceto de la mesa holográfica

4.5.3. Modelos incorporados de creación propia

En este apartado se muestran los modelos incorporados de creación propia capturados dentro de *Substance Painter* una vez texturizados y listos para exportar, o bien dentro de *Unity*.



Figura 21: Nave espacial

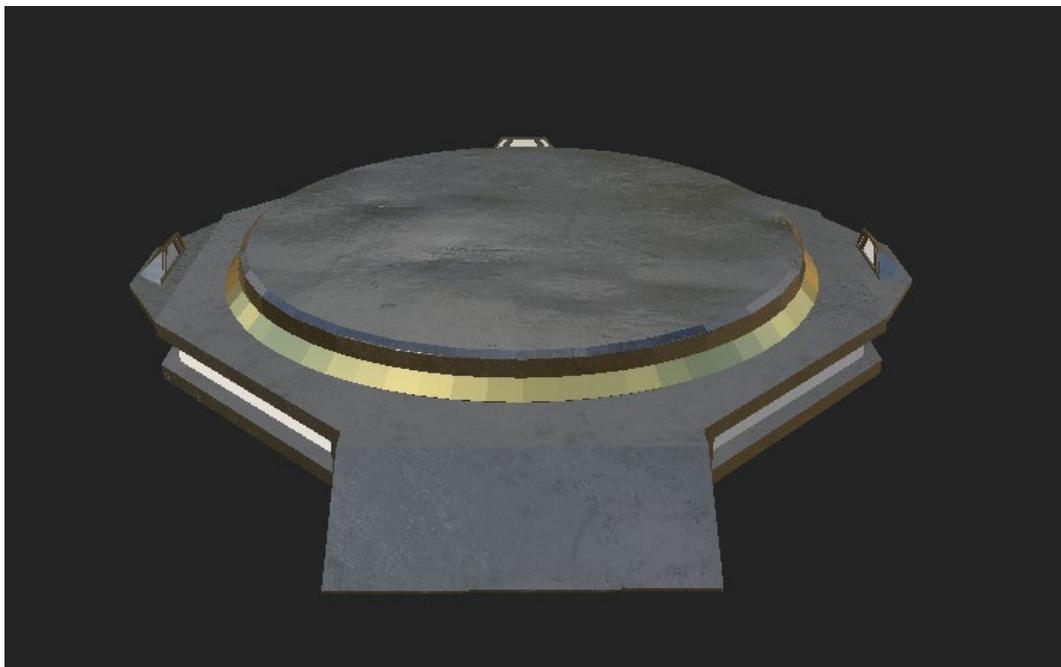


Figura 22: Peana para la nave espacial con rampa



Figura 23: Pilar del boceto de la figura 19 modelado y texturizado

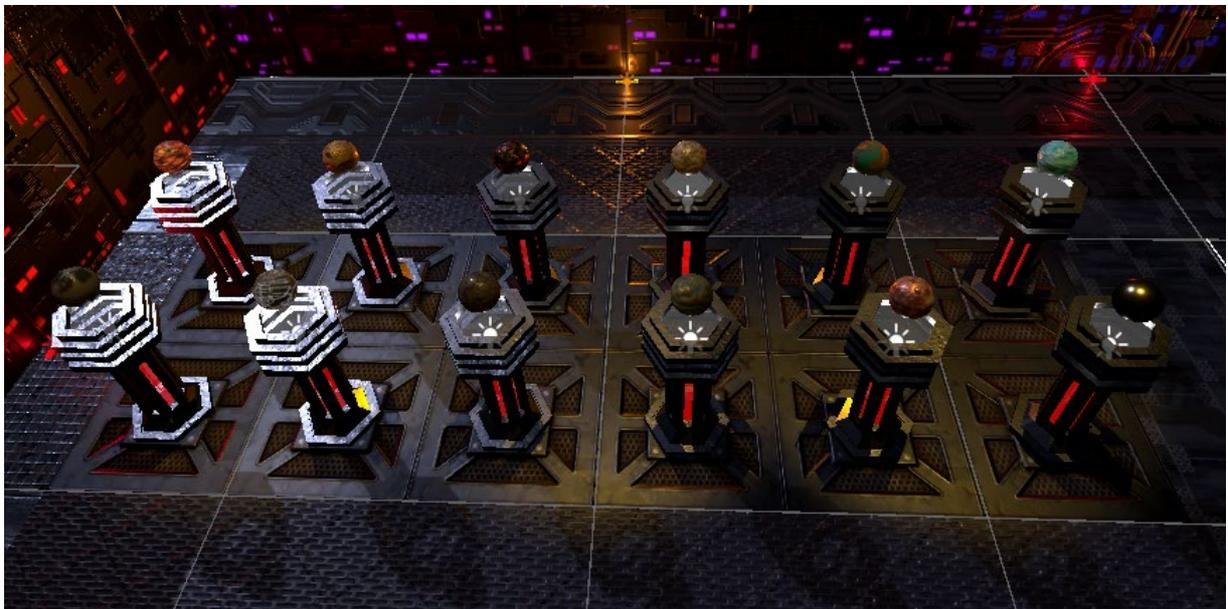


Figura 24: Pilares dentro del nivel exponiendo materiales extraños

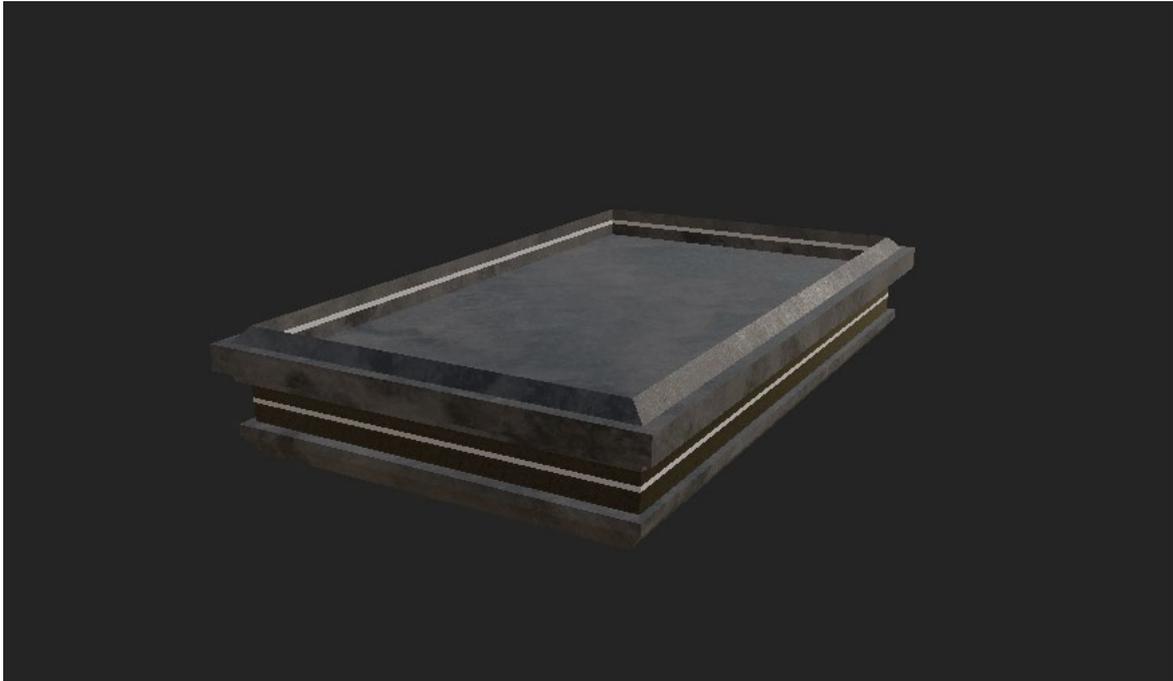


Figura 25: Mesa del boceto de la figura 20 modelado y texturizado

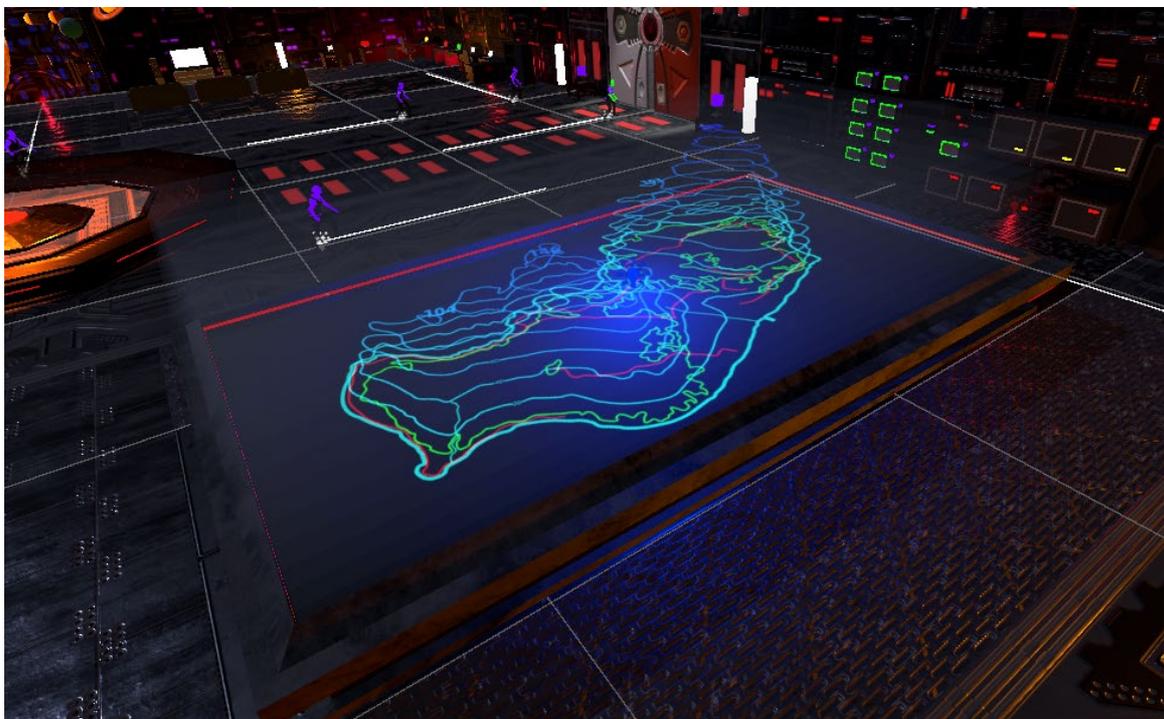


Figura 26: Mesa de la figura anterior dentro del nivel en Unity con un mapa topográfico encima

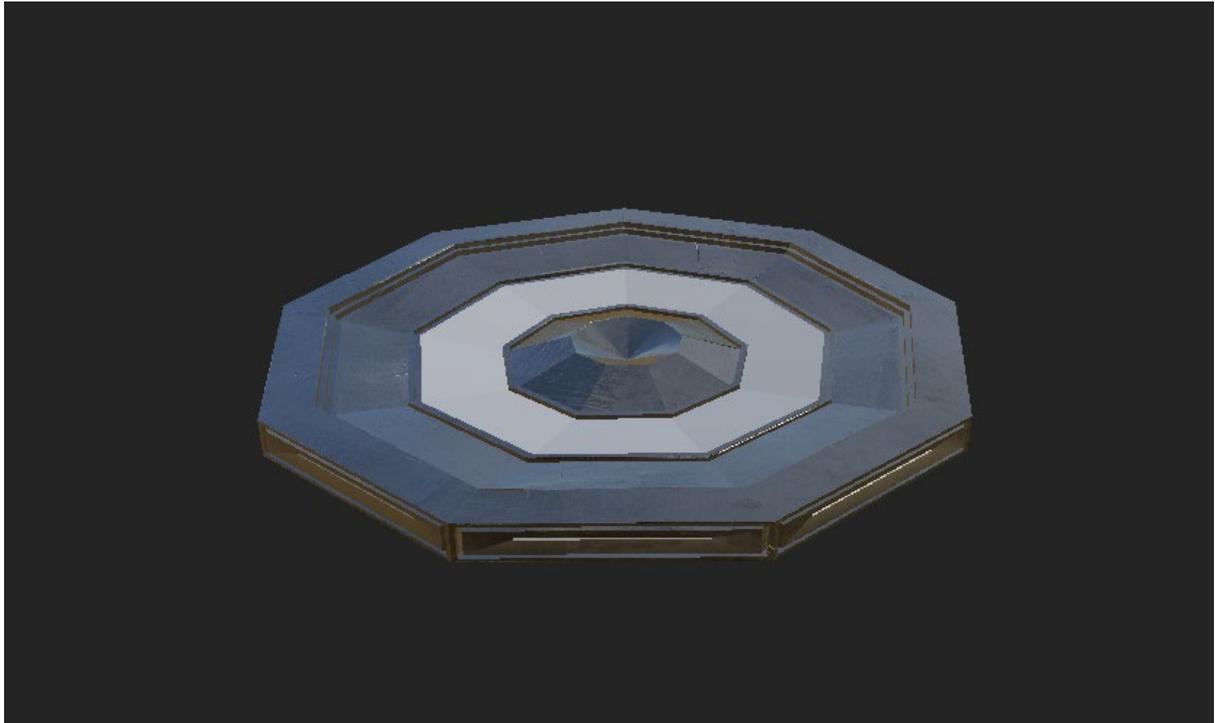


Figura 27: Peana solar

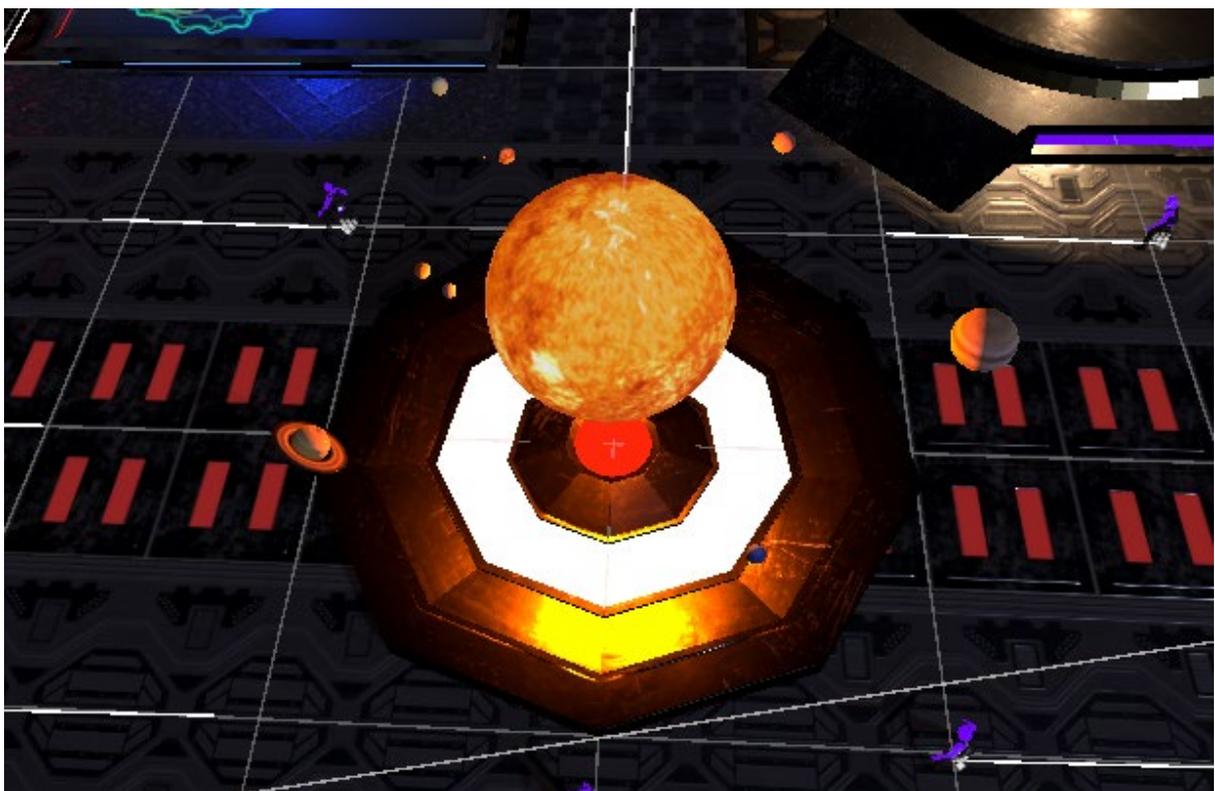


Figura 28: Peana solar dentro del nivel con los planetas y sus texturas descargados de *Adobe Stock*

4.5.4. Modelos y texturas usados de terceros

Para el juego se han usado algunos de los modelos que venían en el *asset* de *Top Down Engine*, Fueron de gran utilidad ya que eran *prefabs* con modelos, texturas y animaciones, además de los scripts y componentes necesarios para ser elementos interactivos funcionales.

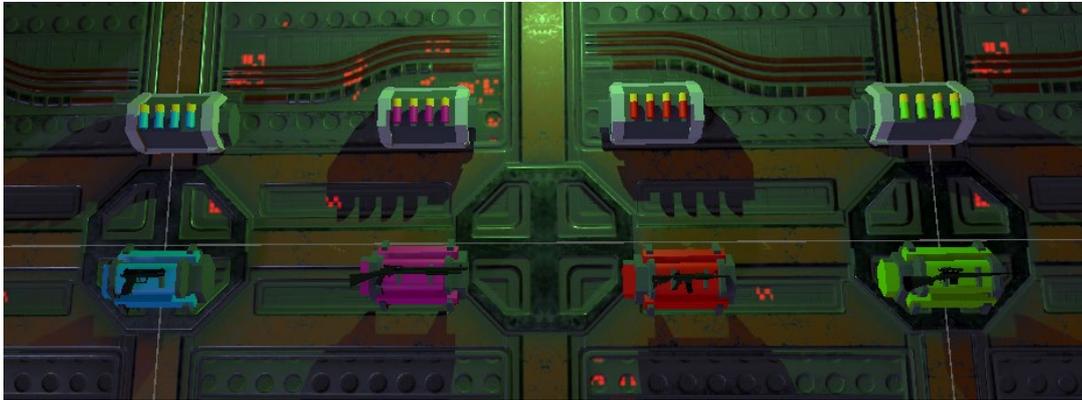


Figura 29: Cajas de armas y municiones



Figura 30: Esferas de vida

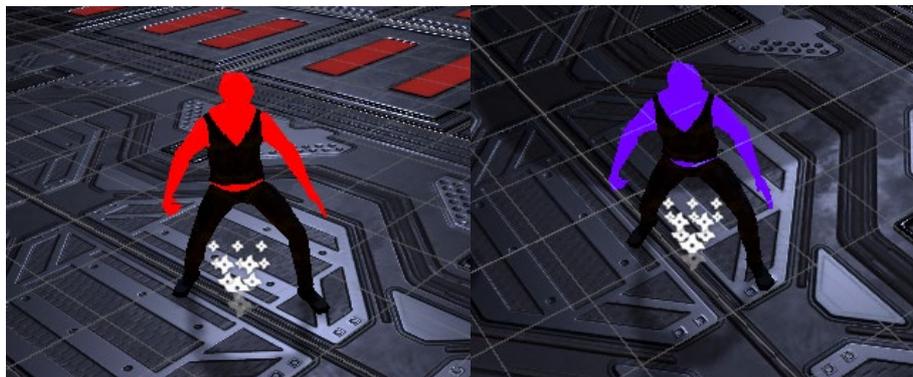


Figura 31: Enemigo de 50 puntos de vida

Figura 32: Enemigo de 200



Figura 33: Personaje jugable

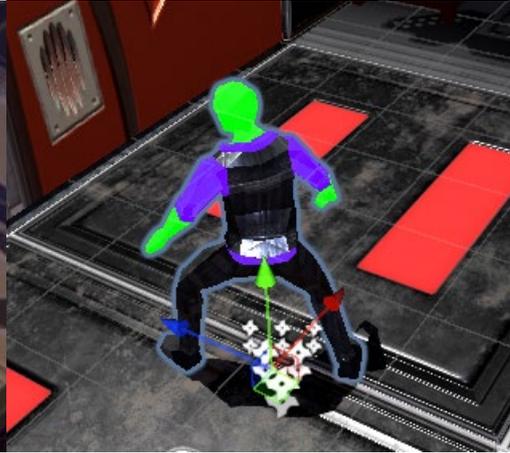


Figura 34: Enemigo/jefe con 1000 puntos de vida

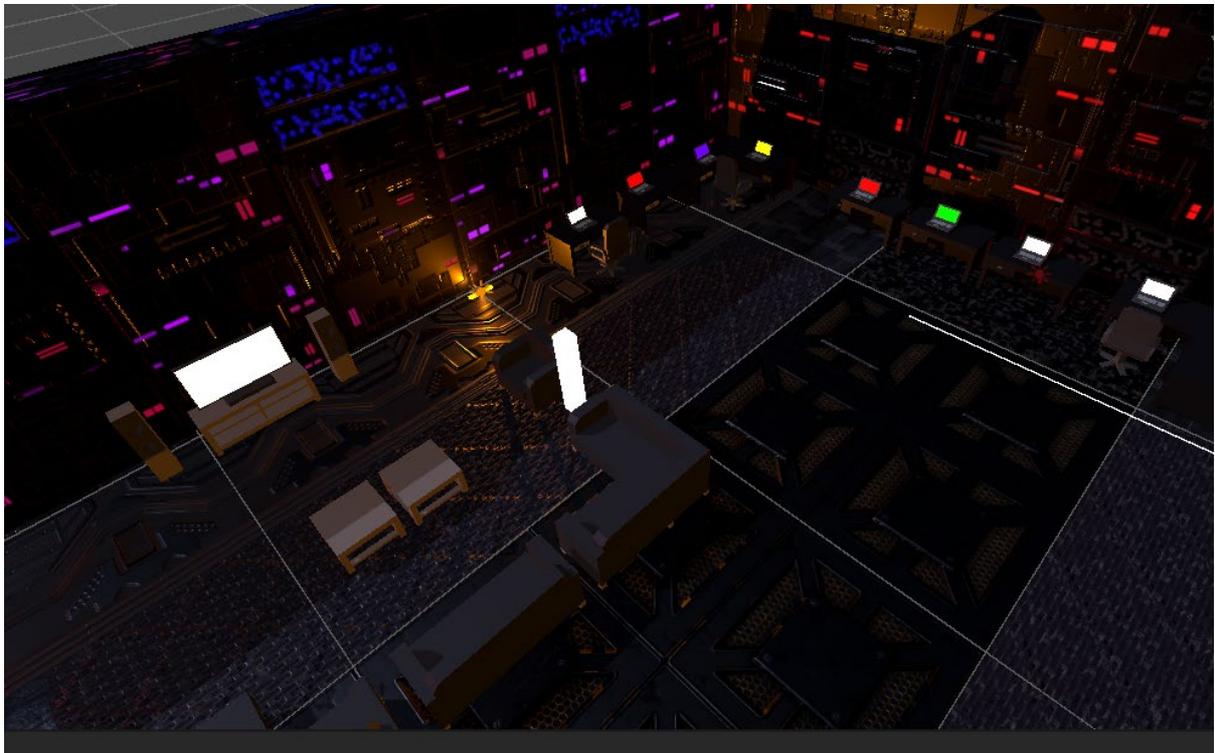


Figura 35: Mesas, altavoces, sillones, portátiles y pantallas del asset "Top Down Engine"

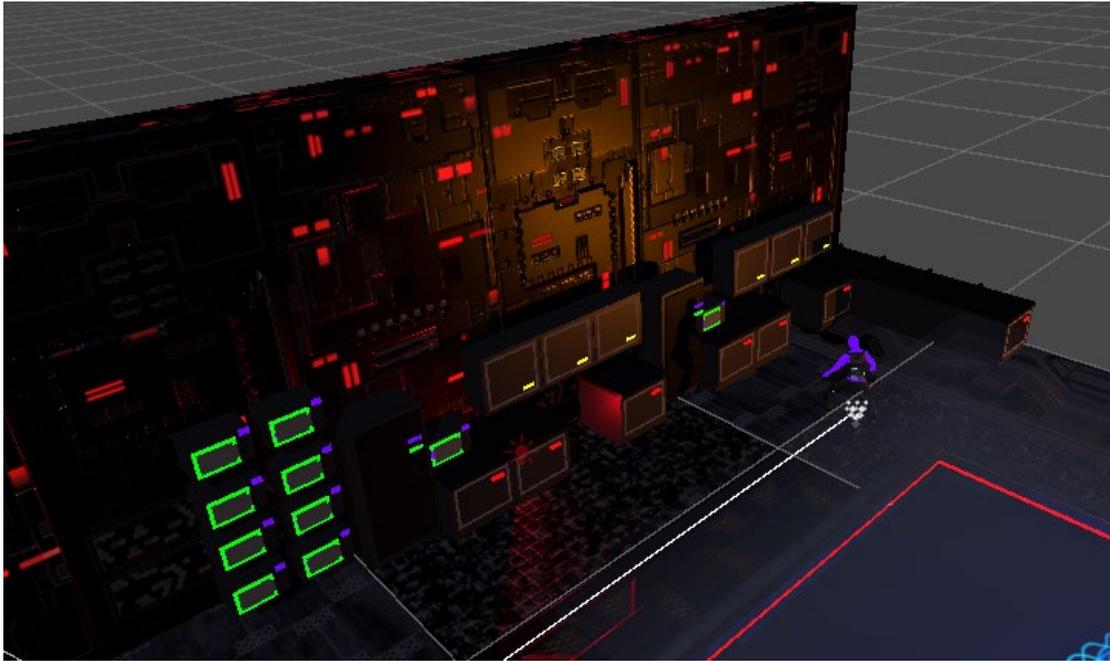


Figura 36: Muebles modificados del asset "Top Down Engine" con materiales luminosos y metálicos



Figura 37: Muebles modificados del asset "Top Down Engine" con un material emisor con texto luminoso

Por otro lado, hay que comentar que todas las texturas se han descargado por medio de adobe Stock, concretamente de la página de la comunidad de *Substance Painter, Community Assets*. También se usaron algunas incluidas en el programa de *Substance* por defecto.

4.5.5. Elementos interactivos

Los elementos interactivos que forman parte del nivel, los podríamos dividir en:

- **Básicos:** suelos paredes y objetos con poco más que un componente *collider* para detectar los bordes.
- **Destruibles:** estos serían todos aquellos que además de las funciones básicas poseen la habilidad de desaparecer de la escena con animaciones y efectos que enfatizan la acción. Pueden o no dejar un objeto al destruirse. Por ejemplo, las puertas no lo hacen, pero los cubos misteriosos si.
- **Destruibles con IA/enemigos:** poseen las características de los anteriores, además de que su animación es mucho más compleja y poseen IA que les dirigen en sus acciones.
- **Recolectables y usables:** tanto los elementos destruibles como los destruibles con IA dan objetos como recompensa. Estos son armas, municiones y orbes de vida, todos ellos están contemplados en las imágenes del subapartado anterior 4.5.4.

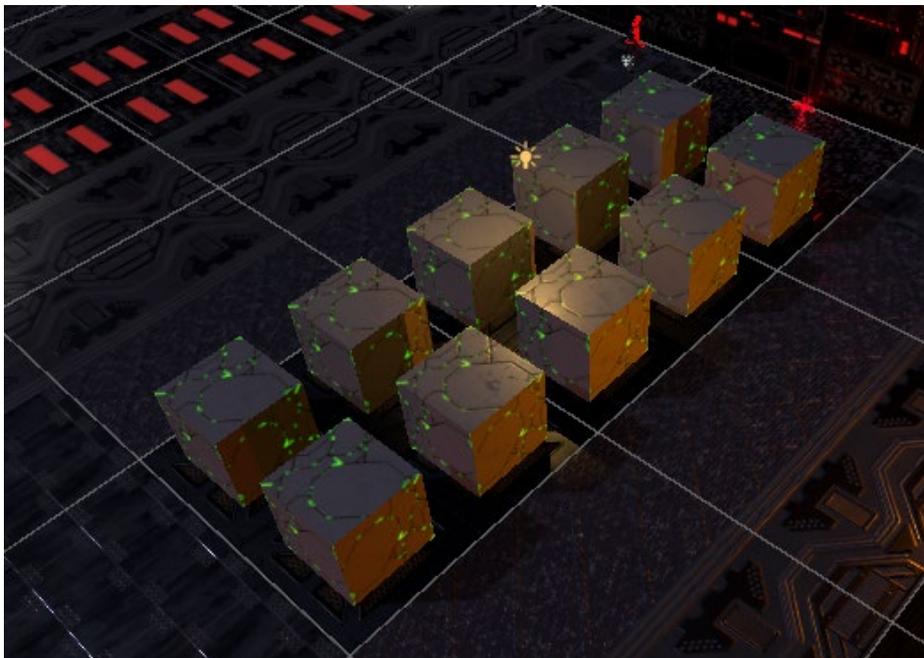


Figura 38: Cubos destruibles con objetos aleatorios

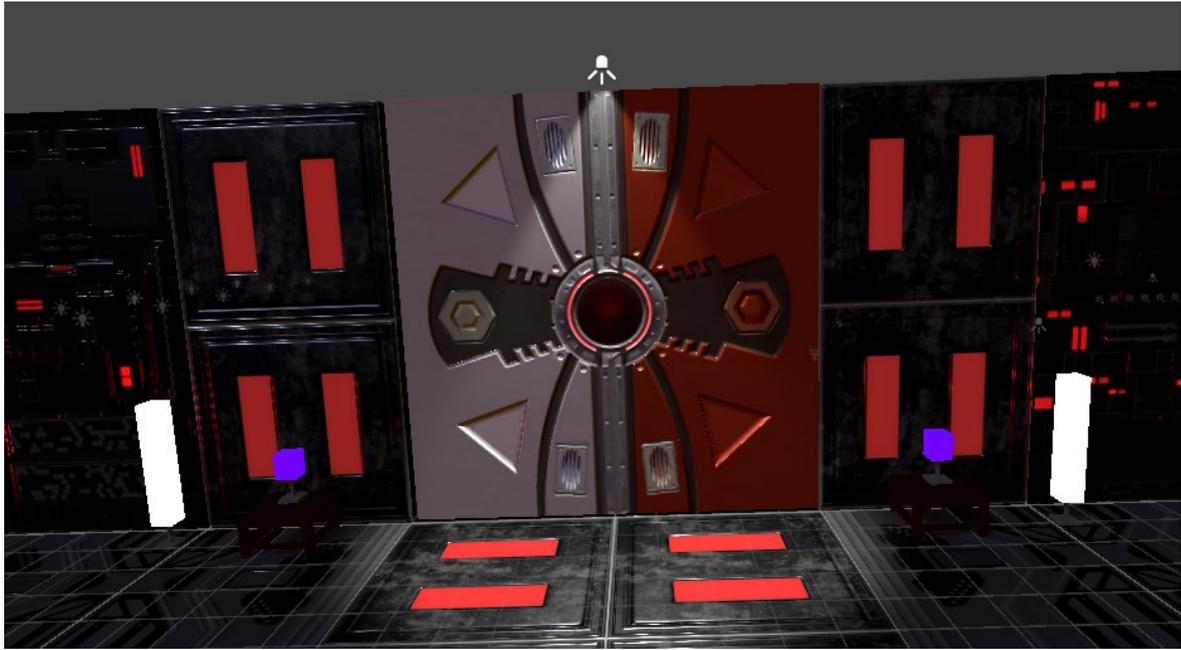


Figura 39: Puertas (destruibles)

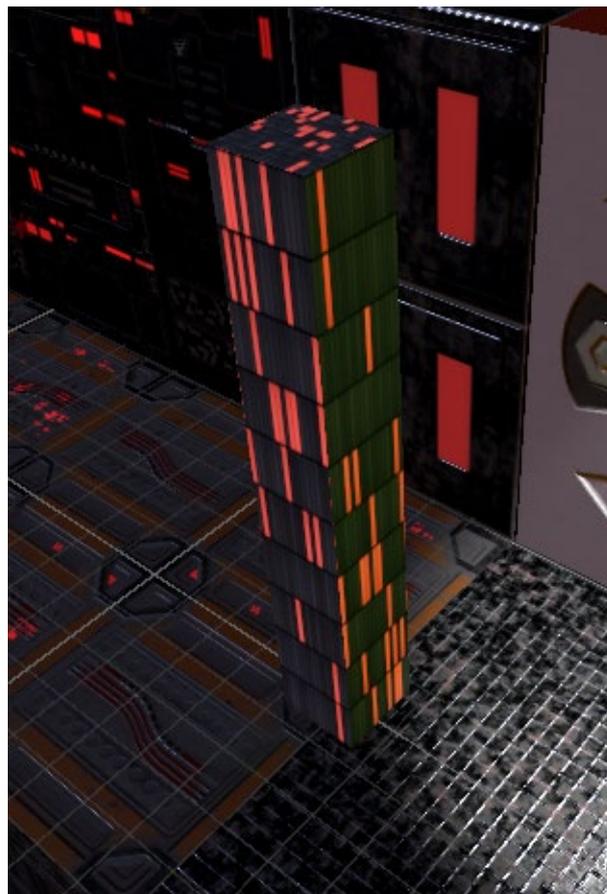


Figura 40: Pilares (destruibles)

5. Implementación

En este apartado se detalla en que tipo de equipos se puede ejecutar el *software* de videojuego "METAL Fury", cuáles son las recomendaciones técnicas de estos y la manera de ejecutarlo.

5.1. Requisitos técnicos recomendados

Para ejecutar el *software* se necesita un mando de Xbox y un PC que posea las siguientes especificaciones :

- SO: Windows 10 o superior.
- CPU: 8 núcleos o superior
- GPU: AMD Radeon o NVIDIA a 8 GB
- RAM: 8 GB RAM
- DirectX 11
- 6 GB de espacio libre

5.2. Instrucciones de ejecución

En este caso, el juego es un *Stand Alone*, por lo que tan solo deberemos de escoger donde queremos descomprimir el ZIP que contiene el código fuente más el ejecutable. Una vez extraído, doble clic sobre el ejecutable para que comience el programa.



Figura 41: Archivos de código fuente y ejecutables

6. Demostración

A continuación, se ofrecen las indicaciones necesarias para entender las mecánicas del juego y el funcionamiento de este.

6.1. Ejemplo de uso del producto o guía de usuario

Una vez hemos abierto el juego, tras la pantalla de carga, este nos lleva al menú principal, donde el jugador deberá pulsar el botón (A) para entrar en el selector de niveles (para más info, ver figura 14 flujo de pantallas). Por el momento, solo existe un nivel de prueba y el resto de las tarjetas son de muestra. Por lo tanto, pulsamos (A) para entrar dentro del nivel 1.

Una vez dentro del nivel, se deben de conocer las mecánicas de juego, las cuales son básicamente destruir el entorno y los enemigos, recoger objetos que dejan como recompensa y usarlos en beneficio del jugador. Si bien en esta primera versión no se ha implementado, la durabilidad de las armas es una propiedad que se piensa implementar.

Para empezar a controlar al personaje, se recomienda pulsar el botón de pausa del juego en donde se ha incluido un diagrama de uso del mando controlador de Xbox (Figura 15).Una vez familiarizado con los controles los elementos interactivos que forman parte de las mecánicas son:

- **Panel Informativo:** estos paneles resaltados con iluminación llamativa y una gran pantalla ofrecen mensajes guía para el jugador.
- **Armas:** Existen 4 tipos de armas con sus municiones (Pistola, rifle de asalto, rifle francotirador y escopeta). Cada una reparte una cantidad de daño distinta, así como también tiene una dispersión y cadencia de tiro variada. El jugador deberá elegir el arma más apropiada para cada cometido.
- **Recolectables:**
 - Cajas de arma y de munición: nos dan arma o municiones.
 - Esferas de vidas: dan vida al tocarlas.
 - Cajas misteriosas: son unos cubos con luz que al destruirlos sueltan objetos aleatorios.
- **Inventario:** El inventario (abrir con el botón del mando *select*) permite almacenar, equipar, desechar y usar objetos. Aquí se guardan los objetos almacenables (ver controles con botón pause o figura 15).
- **Enemigos:** estos intentaran acabar con nuestro personaje tan pronto lo vean aparecer dentro de su rango de detección. Al matarlos sueltan objetos.

- **Coberturas:** el juego posee elementos que permiten esconderse y evadir el daño de los disparos enemigos. Un ejemplo sería los pilares de la entrada.
- **Puertas:** bloquean el acceso y son destruibles. Para ello primero se recomienda eliminar los robots que las vigilan.

Con toda esta información, la misión del jugador es la de sobrevivir y llegar hasta el final del nivel, destruyendo a todos los enemigos y las puertas que se encuentre por su camino.

7. Conclusiones y líneas de futuro

7.1. Conclusiones

En este subapartado se incluyen las conclusiones personales sobre el proyecto derivadas del proceso de trabajo y los resultados obtenidos.

- **Descripción de las conclusiones del trabajo:**
 - ¿Qué lecciones se han aprendido del trabajo?
 - Pese a que existen *assets* que ayudan a elaborar videojuegos, siempre es necesario mucha cantidad de trabajo para crearlos. Tras mi segundo proyecto serio en Unity, veo claro la magnitud de mis intenciones y puedo calibrar mejor mis acciones para llegar a ellas.
 - Modelar y texturizar es un trabajo artesanal que requiere de mucho más tiempo del que imagine. El tema de los mapas UV es muy importante para texturizar apropiadamente.
 - Animar en Unity no es tan sencillo como en *After Effects* o similares, requiere de una programación más abstracta y compleja de dominar, pero lógica y muy útil que ofrece grandes posibilidades.

- **Reflexión crítica sobre el logro de los objetivos planteados inicialmente:**
 - ¿Se han conseguido todos los objetivos? ¿Por qué?
 - Si, pero no del todo. Falta más tiempo para crear un producto sin la ayuda de tantos *assets* con modelos y con una identidad realmente única. Por lo tanto, pese a que el juego si luce un aspecto espectacular, los modelos y texturas no son de creación propia al 100 %.

- **Análisis crítico del seguimiento de la planificación y metodología a lo largo del proyecto:**
 - ¿Se ha seguido la planificación?
 - Si en la gran mayoría del tiempo.
 - ¿La metodología prevista ha sido la adecuada?
 - Si
 - ¿Se han tenido que introducir cambios para garantizar el éxito del trabajo? ¿Por qué?
 - Si. Por falta de tiempo y la carga de trabajo que conlleva realizar los hitos que planteo para este proyecto.

7.2. Líneas de futuro

Como mejoras se espera implementar:

- Incluir una historia.
- Incluir cinemáticas.
- Sistema de niveles del personaje para mejorar sus atributos.
- Crear la propiedad de durabilidad a las armas para usos limitados.
- Sustituir las texturas aplicadas en el juego por texturas de creación propia
- Crear e incluir personajes y elementos del entorno completamente elaborados de cero, incluyendo; modelado, texturizado animaciones de estos.
- Cambiar los efectos de sonido y de las animaciones por otros personalizados.

Bibliografía

Las citas bibliográficas se tienen que ordenar alfabéticamente, según el apellido del autor, o el apellido del primer autor en el caso de ser varios.

1. **10TONS website:** <https://www.10tons.com/> , consultado 11/112022
2. **ASANA website:** <https://asana.com/es/resources/project-management-methodologies>, consultado 11/112022
3. **ASANA website:** <https://asana.com/es/resources/competitive-analysis-example> , consultado 11/112022
4. **GEEKNETIC website:** <https://www.geeknetic.es/PC-Master-Race/que-es-y-para-que-sirve> , consultado 11/112022
5. **FLOW FIRE GAMES website:** <https://www.flowfiregames.com/> , consultado 11/112022
6. **LACED website:**
<https://www.lacedrecords.co/blogs/news/seeing-red-ruiner-s-creative-director-on-how-style-can-precede-substance> , consultado 11/112022
7. **LATAM website:**
<https://www.latamblockchaincouncil.com/news/modelo-de-negocio-play-to-earn-el-puente-entre-los-juegos-y-las-criptomonedas/> , consultado 11/112022
8. **NEGOCIO Y ESTRATÉGIA website:**
<https://negociosyestrategia.com/blog/metodologia-trabajo/#:~:text=Las%20metodolog%C3%ADas%20de%20trabajo%20ayudan,a%20sus%20clientes%20y%20proveedores>, consultado 11/112022
9. **NEUTRÓN website:** <https://neutron.com.ar/> , consultado 11/112022
10. **OBERLO website:** <https://www.oberlo.es/blog/perfil-consumidor> , consultado 11/112022
11. **PULSE OF THE PROGRESION website:**
<https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf>, consultado 11/112022
12. **PricewaterhouseCoopers website:**
<https://www.pwc.com/gx/en/industries/tmt/media/outlook/outlook-perspectives.html#chapter-4> , consultado 11/112022
13. **REIKONGAMES website:** <https://www.reikongames.com/> , consultado 11/112022
14. **RESEARCHGATE website:**
https://www.researchgate.net/publication/261730399_Modelos_de_negocio_emergentes_en_la_industria_del_videojuego , consultado 11/112022
15. **RUINERGAME website:** <http://ruinergame.com/presskit/#description> , consultado 11/112022

16. **SANTANDER BECAS website:**
<https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html> , consultado 11/112022
17. **SIGNIFICADOS website:** <https://www.significados.com/etapas-del-desarrollo-humano/> , consultado 11/112022
18. **STEAM website:** [https://store.steampowered.com/app/979690/The Ascent/](https://store.steampowered.com/app/979690/The_Ascent/) , consultado 11/112022
19. **STEAM website:** [https://store.steampowered.com/app/1594460/I See Red/](https://store.steampowered.com/app/1594460/I_See_Red/) , consultado 11/112022
20. **STEAM website:** <https://store.steampowered.com/app/654050/JYDGE/> , consultado 11/112022
21. **STEAM website:** [https://store.steampowered.com/app/1471410/SYNTHETIK 2/](https://store.steampowered.com/app/1471410/SYNTHETIK_2/) , consultado 11/112022
22. **STEAM website:** <https://store.steampowered.com/app/464060/RUINER/> , consultado 11/112022
23. **Substance painter 3D Community Assets:**
<https://substance3d.adobe.com/community-assets>, consultado 20/11/2022
24. **TRUEARCHIVEMENTS website:** <https://store.steampowered.com/app/464060/RUINER/> , consultado 11/112022
25. **WHITEBOARD GAMES website:** [https://es.wikipedia.org/wiki/Ghost in the Shell](https://es.wikipedia.org/wiki/Ghost_in_the_Shell) , consultado 11/112022
26. **WIKIPEDIA website:** [https://es.wikipedia.org/wiki/Ghost in the Shell](https://es.wikipedia.org/wiki/Ghost_in_the_Shell) , consultado 11/112022
27. **WIKIPEDIA website:** [https://en.wikipedia.org/wiki/Blade Runner](https://en.wikipedia.org/wiki/Blade_Runner) , consultado 11/112022
28. **WIKIPEDIA website:** [https://en.wikipedia.org/wiki/Twin-stick shooter](https://en.wikipedia.org/wiki/Twin-stick_shooter) , consultado 11/112022
29. **WIKIPEDIA website:** [https://en.wikipedia.org/wiki/The Ascent \(video game\)](https://en.wikipedia.org/wiki/The_Ascent_(video_game)) , consultado 11/112022
30. **WIKIPEDIA website:** [https://en.wikipedia.org/wiki/Neon Giant](https://en.wikipedia.org/wiki/Neon_Giant) , consultado 11/112022
31. **WIKIPEDIA website:**
[https://en.wikipedia.org/wiki/Ruiner \(video game\)#Development and release](https://en.wikipedia.org/wiki/Ruiner_(video_game)#Development_and_release) , consultado 11/112022
32. **WIKIPEDIA website:** <https://en.wikipedia.org/wiki/Cyberpunk> , consultado 11/112022

Anexos

Listado de apartados complementarios adicionales o que son demasiado extensos para incluir dentro de la memoria y tienen un carácter auto-contenido. Dependiendo del tipo de trabajo, es posible que no haya que añadir ningún anexo.

Anexo A: Glosario

Glosario de terms y acrónimos utilizados en el trabajo (sólo aquellos mencionados en el presente documento) con breves definiciones de cada uno de ellos.

- *AFX: Audio effect*, efecto de sonido.
- *ARPG: Action Role Player Game*, genero de videojuegos
- *Asset*: término que denomina el contenido multimedia usado para un proyecto
- *Asset Store*: tienda de contenidos multimedia
- *Blender*: programa de modelado y animación en 3D
- *Competitive landscape*: termino que se refiere al panorama competitivo , un diagrama informativo que posiciona a la competencia en los ejes x e y de un mapa de coordenadas para calibrar mejor la circunstancia.
- *Cyberpunk*: subgénero de la ciencia ficción que representa realidades distópicas y apocalípticas en entornos altamente tecnológicos y sofisticados pero con decadencia social y ambiental.
- *IDE: Integrated Development Environment* (IDE), programa de edición de códigos o *scripts*
- *Middleware*: sistema de software que ofrece servicios a las aplicaciones.
- *Moodboard*: tablón con funciones visuales que recopila imágenes inspiracionales para los artistas.
- *SCRUM*: metodología de trabajo ágil.
- *Script*: archivo de texto con códigos de algún lenguaje de programación.
- *TIC*: Técnicas/Tecnologías de la información y la comunicación.
- *Top-Down*: genero de videojuegos caracterizados por la vista cenital en perspectiva isométrica.
- *Photoshop*: programa de adobe para edición de imágenes.
- *PVP: person vs person*
- *PWC* : consultoría multinacional “*PriceWaterhouseCoopers*”
- *Unity*: motor de videojuegos o *real time engine*
- *VFX: Video effects*, efectos de video
- *Visual Studio*: programa IDE para editar código

Anexo B: Entregables del proyecto

Lista de archivos entregados y su descripción.

- Versión Gold Master del juego
- Manual de identidad (1.ª Versión)
- Informe de autoevaluación
- Cesión de derechos TFG

Anexo D: Currículum Vitae

- 2020/2023: Grado de Técnicas de Interacción Digital y Multimedia (UOC, Barcelona, España)
- 2017/2018: Diseñador multimedia (*Immobilien MLM property Brokers*, Torre Vieja, España)
- 2013/2016: Director de eventos (sala *Schmitz Katze, Freiburg im Breisgau, Deutschland*)
- 2010/2012: Titulado en el CFGS de Técnico Superior en Sonido (IES La Guancha, Tenerife, España)
- 2008/2010: Titulado en el CFGM de Laboratorio de Imagen (IES Luis Garcia Berlanga, San Juan, España)