

Boxing Robots

Running Away



Alfonso López Almendros

Grado Multimedia

Videojuegos

Consultora: Ester Arroyo Garriguez

Profesor la asignatura: Joan Arnedo Moreno

Fecha Entrega: 05/06/2020



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>Boxing Robots – Running Away</i>
Nombre del autor:	<i>Alfonso López Almendros</i>
Nombre del consultor/a:	<i>Ester Arroyo Garriguez</i>
Nombre del PRA:	<i>Joan Arnedo Moreno</i>
Fecha de entrega (mm/aaaa):	05/06/2020
Titulación:	<i>Grado Multimedia</i>
Área del Trabajo Final:	<i>Videojuegos</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>TFG, Videojuegos, endless runner</i>
Resumen del Trabajo:	
<p>El presente trabajo, titulado “Boxing Robots – Running Away” representa un producto multimedia de entretenimiento para los jóvenes. Es una elaboración en la que se han incluido diferentes recursos como vídeos, imágenes, sonidos, textos o animaciones, entre otros muchos más, y se han combinado para dar lugar a esta obra única. Una obra enfocada al entretenimiento.</p> <p>Se trata de un videojuego del tipo “Endless Runner”. Un género en el que un personaje avanza constantemente hacia delante evitando una serie de obstáculos y en los que los escenarios, nunca terminan.</p> <p>En este producto, se controla a un robot boxeador que corre hacia delante constantemente huyendo de una gran explosión. La temática es futurista y el juego se desarrolla en un universo imaginario en el que una nueva especie de robots prolifera rápidamente y la humanidad lucha por evitarlo. Para ello, las fábricas o laboratorios en los que construyen estas máquinas están siendo atacadas y destruidas por grupos radicales humanos. En el juego, se encarna a uno de estos robots huyendo de uno de estos laboratorios para salvar la vida.</p> <p>La aplicación se ha desarrollado para PC y Dispositivos Android mediante la plataforma Unity 2019. A su vez, se han utilizado diversas aplicaciones externas para elaborar su contenido. Por ejemplo, se ha recurrido a la suite de Adobe (AfterEffects Illustrator, Photoshop...), a Blender para el modelado 3D o a Audacity para la edición de audio.</p> <p>El resultado es un producto multimedia completo. Un producto de entretenimiento para un público joven que no posee demasiado tiempo para dedicarlo a extensos videojuegos en los que necesitamos emplear muchas horas.</p>	

ABSTRACT (English versión)

The result of this Project is the making of a 3D video game of the Endless Runner type.

The product is designed for Computers running Windows 10 operating system and Android mobile devices.

This document details all the production phases of the product. Looking into detail the different disciplines used for its development related to Multimedia (3D and 2D Design, 3D Modeling, Animations, Audio and Video Treatment, Narrative, Interface Design, and others). On the one hand, the process of preparing or scripting the product is specified, from the conceptualization of this, to the obtaining of the final product. On the other hand, the design processes of both characters, interfaces or scenarios, videos and audios that appear during the game are shown, as well as the formats used, and the software used in each case.

Finally, it shows the use of Unity2019 as a game engine for the final composition of the application and programming using C# in Visual Studio.

The result of the project is a functional and entertaining video game, perfect for evading the mind and having a good time.

ABSTRACT (Versión en español)

El resultado de este proyecto es la realización de un videojuego 3D del tipo Endless Runner.

El producto está diseñado para computadoras con sistema operativo Windows 10 y dispositivos móviles Android.

Este documento detalla todas las fases de producción del producto. Examinando en detalle las diferentes disciplinas utilizadas para su desarrollo relacionadas con multimedia (diseño 3D y 2D, modelado 3D, animaciones, tratamiento de audio y video, narrativa, diseño de interfaz y otros). Por un lado, el proceso de preparación o scripting del producto se especifica, desde la conceptualización de este, hasta la obtención del producto final. Por otro lado, se muestran los procesos de diseño de personajes, interfaces o escenarios, videos y audios que aparecen durante el juego, así como los formatos y el software utilizados en cada caso.

Finalmente, muestra el uso de Unity2019 como motor de juego para la composición final de la aplicación y la programación usando C # en Visual Studio.

El resultado del proyecto es un videojuego funcional y entretenido, perfecto para evadir la mente y pasar un buen rato.

Índice

1. INTRODUCCIÓN	1
1.1 CONTEXTO Y JUSTIFICACIÓN	1
1.2 OBJETIVOS DEL TRABAJO	1
1.3 ENFOQUE Y MÉTODO SEGUIDO	1
1.4 PLANIFICACIÓN DEL TRABAJO	2
<i>Gantt</i> :.....	3
1.5 PRODUCTOS OBTENIDOS	4
1.6 DESCRIPCIÓN DE LOS CAPÍTULOS.....	4
2. ESTADO DEL ARTE	4
2.1. ENDLESS RUNNER	4
1 TER (<i>Un toque</i>).....	5
2 TER (<i>Dos toques</i>).....	5
<i>Otros tipos</i> :	6
2.2. TECNOLOGÍAS DE DESARROLLO:	7
<i>Game Maker Studio 2</i>	7
<i>Construct 3</i>	7
<i>Unity</i>	8
<i>Unreal Engine</i>	10
3. DEFINICIÓN DEL JUEGO	12
3.1. BOXING ROBOTS - RUNNING AWAY	12
3.2. REFERENCIAS	12
3.3 PLATAFORMA E INTERACCIÓN	14
<i>PC</i>	15
<i>Android</i>	15
3.4. CONCEPTUALIZACIÓN	15
3.5. CONCEPT ART	16
<i>Los robots</i>	16
<i>Los escenarios</i>	16
3.6. OBJETIVOS Y MECÁNICAS	17
<i>Movimientos</i> :.....	17
<i>Niveles</i> :	17
<i>Puntos</i> :.....	17
<i>Tiempo</i>	18
<i>Vida</i>	18
<i>Dificultad</i>	18
<i>Velocidad</i>	18
3.7. DESARROLLO - UNITY 3D	18
3.8. ROADMAP	19
1 - <i>Creación del personaje principal</i>	19
2 - <i>Animación</i>	19

<i>Mecánica e interacción</i>	19
4 - <i>Menú</i>	19
5 - <i>Desarrollo del nivel 1</i>	20
6 - <i>Evaluación con usuarios</i>	20
7 - <i>Almacenamiento de resultados</i>	20
8 - <i>Introducción</i>	20
9 - <i>Desarrollo del nivel 2</i>	20
10 - <i>Desarrollo del nivel 3</i>	20
11 - <i>Evaluación con usuarios</i>	20
12 - <i>Retoques finales</i>	20
4. DISEÑO TÉCNICO	20
4.1. ENTORNO DE DESARROLLO – UNITY 2019	20
<i>Paneles</i>	21
<i>Proyectos</i>	24
4.2. REQUISITOS TÉCNICOS	28
<i>Unity Editor</i>	28
4.3. HERRAMIENTAS EMPLEADAS	28
<i>Blender</i>	28
<i>Mixamo</i>	30
<i>Photoshop</i>	31
<i>Illustrator</i>	32
<i>After Effects</i>	33
4.4. INVENTARIO DE RECURSOS	34
<i>RECURSOS Externos:</i>	34
<i>RECURSOS Propios</i>	35
4.5. ARQUITECTURA DEL JUEGO	36
5. DISEÑO DE NIVELES	37
5.1. PRIMERA ESCENA – LA INTRODUCCIÓN	37
5.2. NIVEL 1 – SALIDA DEL LABORATORIO	38
5.3. NIVEL 2 – DESCENDIENDO A LA TIERRA	39
5.4. NIVEL 3 – EN LA TIERRA	40
6. MANUAL DE USUARIO	41
6.1. REQUISITOS TÉCNICOS PARA JUGAR	41
<i>A través de Unity Player para PC</i>	41
<i>dispositivos ANDROID</i>	42
7. CONCLUSIONES	42
8. GLOSARIO	44
9. BIBLIOGRAFÍA	47

Lista de figuras

1 - Captura de pantalla del videojuego Sloppy Bird	5
2 - Captura de pantalla del videojuego Jetpack Joyride	5
3 - Captura de pantalla del videojuego Infinity Runner.....	6
4 - Captura de pantalla del videojuego Infinity Runner.....	6
5 - Imagen de un juego creado por Game Maker Studio 2	7
6 - Imagen de un juego creado por Game Maker Studio 2	8
7 - Imagen de un juego creado por Game Maker Studio 2	8
8 - Imagen del juego Cuphead, creado con Unity	9
9 - Imagen del juego Cuphead, creado con Unity	9
10 - Imagen del juego Cuphead, creado con Unity	9
11 - Imagen del juego Ghost of a Tale, creado con Unity.....	10
12 - Imagen del juego Borderlands, creado con Unreal Engine	11
13 - Imagen del juego Borderlands, creado con Unreal Engine	11
14 - Imagen del juego Borderlands, creado con Unreal Engine	11
15 - Captura del juego Imposible Road.	12
16 - Captura del juego Mario Kart 8.	13
17 - Captura del juego Minion Rush: Gru – Mi Villano favorito.	13
18 - Captura del juego Temple Run.	13
19 - Captura del juego Subway Surfers.	14
20 - Captura del juego Lemmings.....	14
21 – Click and Swipe.	15
22 – Swipe.	15
23 - Boceto del personaje principal. Creado en Illustrator.....	16
24 - Imagen demo de Sci-fi Style Modular Pack.	17
25 – Panel Scene de Unity.	21
26 – Panel Hierarchy de Unity.	22
27 – Panel Project de Unity.	22
28 – Panel Inspector de Unity.	23
29 - Panel Game de Unity.....	23
30 - Panel Console de Unity.....	23
31 - Otros paneles de Unity.....	24
32 – Scripts. Imagen capturada desde el propio software.	25

33 – Código C# de Unity.	26
34 – Prefabs de Unity.	26
35 - Componentes de Unity.....	27
36 - Cámara de Unity.....	27
37 - Modelo 3D del Robot en Blender.....	28
38 - Esqueleto del robot para Mixamo.....	29
39 - Materiales asignados al robot en Blender.....	29
40 - Resultado del modelado 3D del robot en Blender.....	30
41 - Puntos clave de la animación en Mixamo.....	30
42 - Animación "Running" de Mixamo.....	31
43 - Botón Play en Photoshop.....	31
44 - Botón Play en Unity.....	32
45 - imagen para pantallas TV de la introducción en Illustrator.....	32
46 - Animación en After Effects.....	33
47 - Vista de la animación en Unity.....	33
48 - Arquitectura del videojuego.....	36
49 - El escenario de la introducción.....	37
50 - Guantes de robot.....	37
51 - Robots destrozados.....	37
52 - Destrucción del laboratorio.....	38
53 - Planetas Orbitando.....	39
54 - Camino curvado.....	39
55 - Rotación de la tierra.....	40
56 - Nivel 2.....	40
57 - Nivel 3.....	41

1. INTRODUCCIÓN

1.1 CONTEXTO Y JUSTIFICACIÓN

El producto multimedia que presentamos a continuación trata de cubrir una necesidad humana muy importante, el ocio.

Actualmente, las personas apenas tienen tiempo para dedicarlo al ocio y, aun menos, a productos interactivos que requieren de largos periodos de concentración frente a una pantalla. Los videojuegos basados en largas historias cada vez tienen menos relevancia y proliferan más los minijuegos o juegos cortos que podemos disfrutar desde nuestro dispositivo móvil o tableta.

Sin embargo, no sólo hay que centrarse en estos dispositivos. Muchas personas pasan la mayor parte de su vida frente a un ordenador y, dedicarle unos minutos a un videojuego en ese mismo dispositivo, satisfaría notablemente su necesidad de ocio.

Hoy en día, las empresas buscan un equilibrio entre la carga de trabajo y el bienestar de los trabajadores. Fomentar el ocio puede contribuir notablemente a esta labor. Y si, además, podemos hacerlo sin que las personas tengan que moverse de su puesto de trabajo, estaremos contribuyendo aún más al rendimiento de las personas y a su bienestar laboral. Por ello, he creado un videojuego compatible con ambas plataformas.

1.2 OBJETIVOS DEL TRABAJO

El principal objetivo del trabajo es la creación de un producto multimedia funcional enfocado en el ocio de las personas.

Se trata de crear un videojuego entretenido, adictivo y de corta duración que permita entretenerse en cortos periodos de tiempo.

1.3 ENFOQUE Y MÉTODO SEGUIDO

La idea de este proyecto es la creación de un producto nuevo e innovador, basado en un tipo de videojuegos ya existente, pero con un enfoque personal.

Para ello, he intentado contar una historia de invención propia durante el juego. He creado un personaje principal (El robot), le he asignado un problema (quieren destruirlo) y la forma de solucionarlo (huir).

Así, adaptando todo esto al tipo de videojuego escogido, podemos crear perfectamente el guion que seguiremos para su creación y obtener un producto único de gran valor.

1.4 PLANIFICACIÓN DEL TRABAJO

Una vez la idea del videojuego es clara, he tenido que darle forma creando el documento de diseño del videojuego (GDD). Aquí se recopila la idea, la conceptualización y el *Roadmap* a seguir durante la elaboración de este. El documento se definió para el 1 de marzo.

La elaboración del proyecto la llevé a cabo a partir del 2 de marzo y dividí el esfuerzo en tres bloques:

1. El primer esfuerzo conlleva la realización del personaje principal, la creación de sus animaciones, la mecánica de interacción con el mismo, la creación de los menús del juego y la creación de un primer nivel funcional sin entrar en detalles estéticos.

Para llevar a cabo esta primera parte, utilizaremos *Blender* para la creación del personaje y *Mixamo* para las animaciones de este. Posteriormente, utilizaremos *Unity* para la creación del nivel, menús e ir componiendo los distintos elementos del juego. La fecha final de esta etapa es el 5 de abril.

2. La segunda etapa comprende el almacenamiento de las principales variables del juego en el equipo para futuras partidas, 3 niveles completos y la creación de la introducción.

Continuamos utilizando el software *Unity* para la implementación de todos los elementos del juego y la creación de niveles. También utilizaremos diferentes herramientas de la suite de *Adobe* (*Photoshop*, *Illustrator*, *AfterEffects* y *Audition*) para crear *assets* o elementos que añadimos al juego. La fecha límite para la realización de este esfuerzo es el 24 de mayo.

3. La última etapa comprende la modificación respecto al *feedback* de los usuarios de prueba, retoques finales y la compilación de la aplicación.

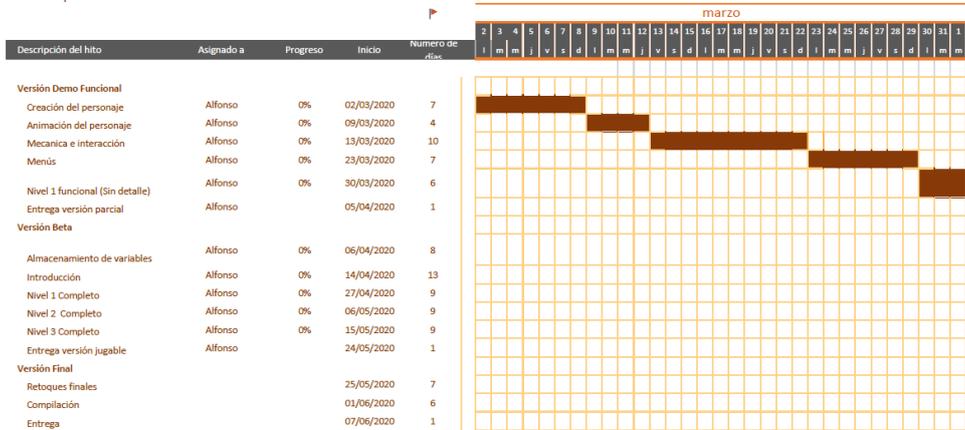
GANTT:

[3] Incluimos a continuación el diagrama de Gantt dividido por meses:

MARZO

Boxing Robots - Running away

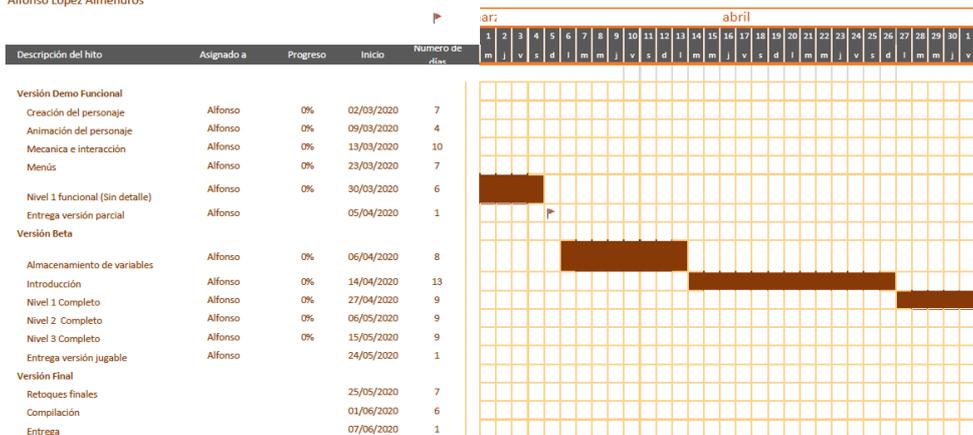
UOC - Grado Multimedia Fecha de inicio del proyecto: 02/03/2020
TFG - Videjuegos
Alfonso López Almendros



ABRIL

Boxing Robots - Running away

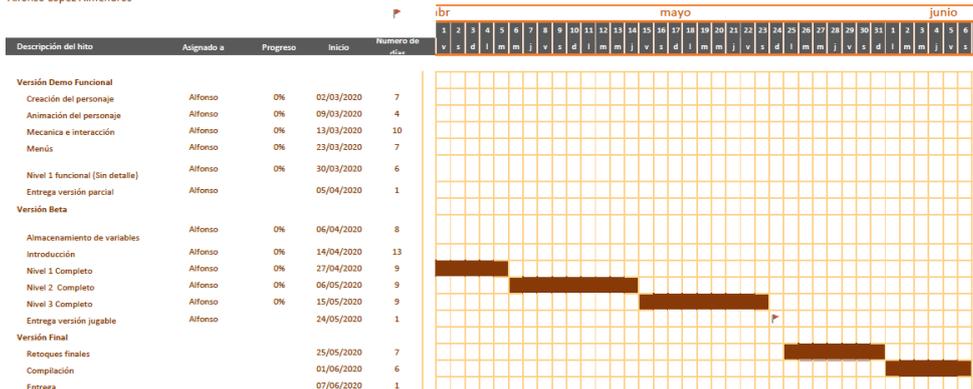
UOC - Grado Multimedia Fecha de inicio del proyecto: 02/03/2020
TFG - Videjuegos
Alfonso López Almendros



MAYO Y JUNIO

Boxing Robots - Running away

UOC - Grado Multimedia Fecha de inicio del proyecto: 02/03/2020
TFG - Videjuegos
Alfonso López Almendros



1.5 PRODUCTOS OBTENIDOS

El resultado de del proyecto es un producto multimedia totalmente funcional orientado al ocio, que podremos ejecutar desde cualquier PC con Windows o desde un dispositivo Android como un móvil o una tableta.

1.6 DESCRIPCIÓN DE LOS CAPÍTULOS

En los siguientes capítulos podemos encontrar toda la información en detalle del proyecto.

- El estado del arte, dónde se analizan algunos videojuegos similares existentes en el mercado, así como las diferentes tecnologías de desarrollo que se utilizan en la actualidad.
- La definición del juego, dónde encontramos una información más ampliada sobre el diseño y la conceptualización del este, el *Concept Art* y el *roadmap*.
- El diseño técnico, dónde se especifican las herramientas utilizadas y los requisitos de la aplicación, así como los recursos utilizados.
- El diseño de niveles, dónde se analiza el diseño de cada uno de los niveles.
- Y, por último, el manual de usuario, dónde se describe cómo utilizar el producto.

2. ESTADO DEL ARTE

2.1. ENDLESS RUNNER

[1] Esencialmente, el género “Endless runner” se refiere a aquellos juegos que nunca terminan y en los que el personaje se desplaza forzosamente hacia delante y, cuanto más avanzamos, mayor es la velocidad y en ende la dificultad para superar los obstáculos que aparecen por el camino.

[2] Es un género que resulta de la adaptación de los videojuegos de plataformas 2D que existían en los años 90, a las nuevas plataformas tales como tabletas o teléfonos móviles.

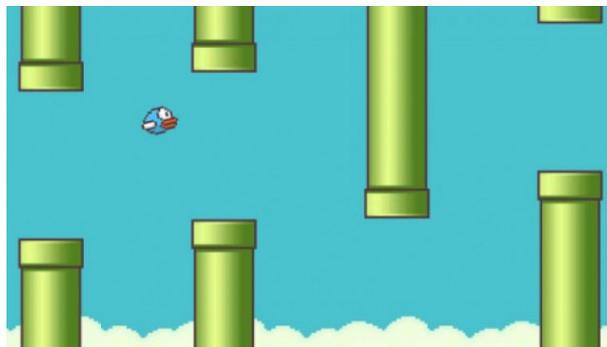
La forma de jugar ha cambiado y, con estos dispositivos, ya no se buscan videojuegos en los que pasamos varias horas pegados a un monitor. Ahora, son videojuegos de escasos minutos que buscan atrapar al jugador lo máximo posible.

Por otro lado, los controles se simplifican para ofrecer una mejor experiencia de juego en estos nuevos dispositivos y, en este caso, por ejemplo, se programa el personaje principal para que avance automáticamente y, así, evitar el botón de avanzar y dejar más posibilidades de movimiento con menos botones o interacciones.

Al principio surgieron varios subgéneros en función del control que hacemos del personaje:

1 TER (Un toque)

Sólo existe un control táctil con el que controlamos el salto o la altura del personaje. Simplemente tocamos la pantalla para movernos por el nivel. Por ejemplo: *Sloppy Bird*



1 - Captura de pantalla del videojuego Sloppy Bird

[11/03/2020] [ONLINE] Extraído de: <<https://play.google.com/store/apps/details?id=id.beeper.sloppybird>>

2 TER (DOS TOQUES)

El personaje avanza automáticamente y, únicamente podemos saltar y disparar. Por ejemplo: *Jetpack Joyride*



2 - Captura de pantalla del videojuego Jetpack Joyride

[11/03/2020] [ONLINE] Extraído de: <<https://play.google.com/store/apps/details?id=com.halfbrick.jetpackjoyride>>

OTROS TIPOS:

Actualmente, con la rápida evolución de los dispositivos móviles, existen más tipos de “*Endless Runner*” en los que podemos manejar al personaje deslizando los dedos por la pantalla o con sensores de movimiento (Acelerómetros o giroscopios). También han evolucionado del 2D al 3D creando juegos más vistosos y atractivos para los usuarios actuales.

Por otro lado, estos juegos también han evolucionado para otras plataformas como PCs o videoconsolas con un gran número de funcionalidades y gráficos muy avanzados.

Por ejemplo: *Infinity Runner* (PS4)



3 - Captura de pantalla del videojuego *Infinity Runner*

[25/03/2020] [ONLINE] Extraído de: <<https://www.playstation.com/en-us/games/infinity-runner-ps4/>>



4 - Captura de pantalla del videojuego *Infinity Runner*

[25/03/2020] [ONLINE] Extraído de: <<https://www.playstation.com/en-us/games/infinity-runner-ps4/>>

2.2. TECNOLOGÍAS DE DESARROLLO:

Actualmente existen diversas plataformas para el desarrollo de videojuegos, o game engines, tanto en 2D como en 3D. Muchas de ellas incluyen versiones gratuitas con las que inicializarse en este sector. Las más utilizadas para la realización de este tipo de videojuegos son:

GAME MAKER STUDIO 2

[4] Motor multiplataforma para el desarrollo de videojuegos 2D en el que apenas es necesario programar. Utiliza un lenguaje de programación propio llamado Game Maker Lenguaje (GMK), y su interfaz principal, utiliza el sistema de “Arrastrar y soltar “para una programación más intuitiva y sencilla para el programador.



5 - Imagen de un juego creado por Game Maker Studio 2

[25/03/2020] [ONLINE] Extraído de:< <https://www.indiedb.com/engines/gamemaker-studio-2/images/banner-auto-tiles-5#imagebox> >

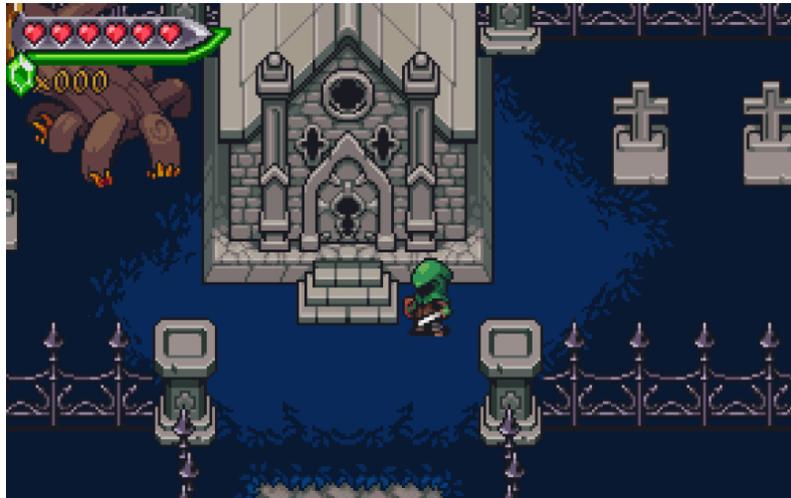
Posee una gran cantidad de recursos disponibles para utilizar en nuestros proyectos y una amplia comunidad de desarrolladores.

Existe una versión de prueba gratuita de 30 días para Windows y Mac, también es posible adquirir varios tipos de licencias dependiendo de la finalidad del juego a realizar, así como una versión educativa para estudiantes.

CONSTRUCT 3

[5] Motor multiplataforma de entorno web y programado en HTML5. De fácil manejo y con una amplia cantidad de recursos de aprendizaje.

En este caso, la programación se realiza mediante las denominadas “Hojas de eventos”. Estas hojas contienen una serie de declaraciones, condicionales o disparadores para realizar determinadas funciones o acciones.



6 - Imagen de un juego creado por Game Maker Studio 2

[25/03/2020] [ONLINE] Extraído de: < <https://www.indiedb.com/engines/gamemaker-studio-2/images/banner-auto-tiles-5#imagebox> >



7 - Imagen de un juego creado por Game Maker Studio 2

[25/03/2020] [ONLINE] Extraído de: < <https://www.indiedb.com/engines/gamemaker-studio-2/images/banner-auto-tiles-5#imagebox> >

Está disponible gratuitamente con limitaciones para el uso personal.

UNITY

[6] Plataforma para el desarrollo de videojuegos multiplataforma tanto en 2D como en 3D en tiempo real. Completamente compatible con los principales softwares de diseño

del mercado como Blender, 3Ds Max, Maya, ZBrush, Photoshop, etc... así como multitud de plataformas de desarrollo mediante plugins de terceros.



8 - Imagen del juego Cuphead, creado con Unity

[25/03/2020] [ONLINE] Extraído de: < <https://unity.com/es/madewith/cuphead>>



9 - Imagen del juego Cuphead, creado con Unity

[25/03/2020] [ONLINE] Extraído de: < <https://unity.com/es/madewith/cuphead>>



10 - Imagen del juego Cuphead, creado con Unity

[25/03/2020] [ONLINE] Extraído de: < <https://unity.com/es/madewith/cuphead>>

La programación se realiza mediante en C Sharp o javascript utilizando Microsoft Visual Studio actualmente para la depuración de scripts.

Por otro lado, dispone de potentes herramientas para Animacion (Mecanim), creación de escenas, iluminación, render, efectos, IA, etc... que hacen de esta herramienta una de las más potentes en el mercado.



11 - Imagen del juego Ghost of a Tale, creado con Unity

[25/03/2020] [ONLINE]Extraído de: < <https://unity.com/es/madewith/ghost-of-a-tale>>

Dispone de una amplia comunidad de desarrolladores, así como una gran cantidad de recursos para utilizar en nuestros proyectos.

Para el uso de Unity, existe una versión personal gratuita con algunas restricciones y un máximo de ingresos por la publicación de los desarrollos.

UNREAL ENGINE

[7] Plataforma de desarrollo de videojuegos 3D en tiempo real para la realización de videojuegos de última generación. Al igual que Unity, dispone de una gran compatibilidad con aplicaciones de terceros para diseño 3D, Audio, XR, etc...

Al igual que la mencionada anteriormente, Unreal integra herramientas para la animación, creación de mundos, rendering, iluminación, materiales, etc...



12 - Imagen del juego Borderlands, creado con Unreal Engine

[25/03/2020] [ONLINE] Extraído de: < <https://borderlands.com/es-ES/picture#picture-p=3>>



13 - Imagen del juego Borderlands, creado con Unreal Engine

[25/03/2020] [ONLINE] Extraído de: < <https://borderlands.com/es-ES/picture#picture-p=3>>



14 - Imagen del juego Borderlands, creado con Unreal Engine

[25/03/2020] [ONLINE] Extraído de: < <https://borderlands.com/es-ES/picture#picture-p=3>>

Es de uso gratuito y únicamente nos cobrará un 5% cuando nuestro producto tenga éxito. Sin restricciones en el uso.

3. DEFINICIÓN DEL JUEGO

3.1. BOXING ROBOTS - RUNNING AWAY

Juego del género “Endless runner” en el que nuestro personaje principal, un robot boxeador futurista, avanza sin remedio hacia delante huyendo de una gran explosión.

Los movimientos de este nos permitirán esquivar todo tipo de objetos que encontraremos en la huida, saltando, rodando, o desplazándonos de izquierda a derecha.

El objetivo principal será llegar a la zona segura antes de que el tiempo se acabe nos quedemos sin vida, por lo que tendremos que intentar evitar el mayor número de obstáculos durante la huida.

Cada vez que alcancemos la zona segura, desbloquearemos un nuevo nivel. Los dos primeros niveles serán de preparación para jugar al tercer nivel. En este, no habrá fin y tendremos que conseguir llegar lo más lejos posible para obtener la máxima puntuación.

3.2. REFERENCIAS

La idea para la realización del camino a seguir surge de títulos como “Impossible Road” o “Mario Kart (Nivel del espacio)” en el que avanzamos sin nada a nuestro alrededor.



15 - Captura del juego Impossible Road.

En Wonderfull Lasers, [Online] <http://www.wonderful-lasers.com/>



16 - Captura del juego Mario Kart 8.

En Nintendo, [Online] <https://www.nintendo.es/Juegos/Wii-U/Mario-Kart-8-765384.html>

La mecánica del movimiento la he ideado basándome en juegos como “Minion Rush: Gru – Mi Villano Favorito”, “Temple Run” o “Subway Surfers” en los que el jugador dispone de 3 carriles a los que desplazarse para evitar obstáculos.



17 - Captura del juego Minion Rush: Gru – Mi Villano favorito.

En Gameloft [Online] <https://www.gameloft.com/en/game/despicable-me-2>



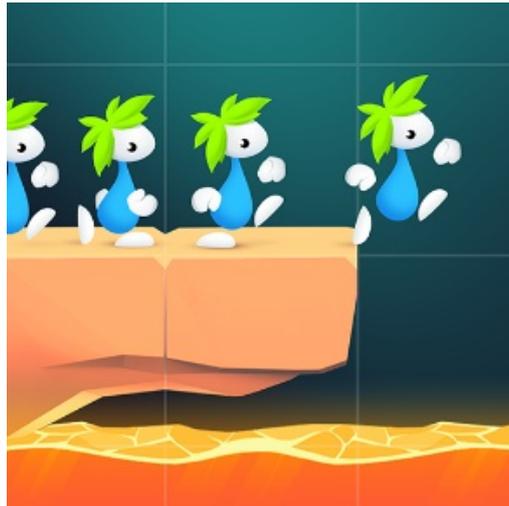
18 - Captura del juego Temple Run.

En Imangistudios. [Online] <https://www.imangistudios.com/games.html>



19 - Captura del juego Subway Surfers.
En Sybogames [online] <https://sybogames.com/>

Además, nuestro jugador no avanzará solo, sino que otros muchos robots boxeadores nos acompañarán a nuestro alrededor huyendo de la explosión e iremos viéndolos chocar, caer o esquivar obstáculos a nuestro alrededor al estilo “Lemmings”.



20 - Captura del juego Lemmings.
En Sadpuppy. [Online] <https://sadbuppy.games/games/>

3.3 PLATAFORMA E INTERACCIÓN

El juego se desarrollará con Unity y se compilará para PC como aplicación de escritorio y para Android para su disfrute en dispositivos móviles o tabletas.

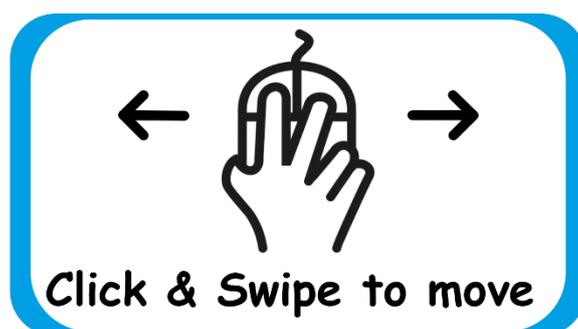
PC

[8] El control se realiza mediante el ratón.

Nos moveremos por lo menús pulsando con el botón izquierdo del ratón sobre los correspondientes botones.

Para mover al personaje durante el juego, pulsamos con el botón izquierdo del ratón en cualquier parte de la pantalla, arrastramos el ratón en una dirección y soltamos el botón.

Si deslizamos hacia arriba, el personaje saltará, si lo hacemos hacia abajo, el personaje rodará, y de lado, cambiaremos de carril.



21 – Click and Swipe.

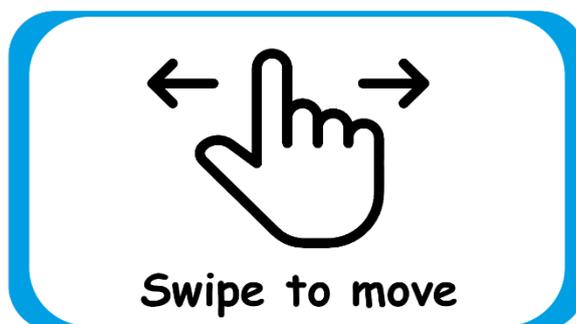
Imagen creada para el videojuego

ANDROID

[8] El control se realiza de forma táctil, es decir, tocando la pantalla con los dedos.

Nos desplazamos por los menús tocando los botones correspondientes en cada caso y moveremos al personaje deslizando el dedo por la pantalla.

Igualmente, si deslizamos el dedo hacia arriba el personaje saltará, si lo hacemos hacia abajo rodará y si lo hacemos de lado, el personaje cambiará de carril.



22 – Swipe.

Imagen creada para el videojuego

3.4. CONCEPTUALIZACIÓN

En un mundo futurista, existen laboratorios en el espacio exterior, donde se fabrican multitud de robots boxeadores (deporte de moda en esa época). Sin embargo, un grupo de humanos revolucionarios en contra de esta fabricación desmesurada, provocan explosiones en las instalaciones para eliminar a todos estos nuevos robots, por lo que estos, huyen rápidamente intentado no ser alcanzados por las explosiones.

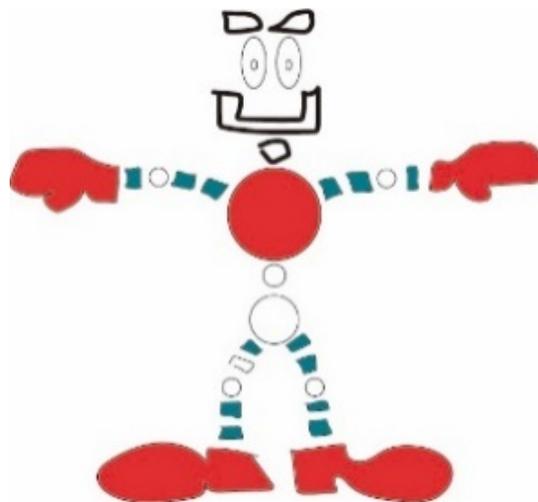
Nosotros nos ponemos en el papel de uno de estos robots y, nuestro objetivo, al igual que el de los demás robots, es llegar a un lugar seguro y no ser alcanzados por la explosión.

En cada nivel, tendremos que huir durante un determinado tiempo para no ser alcanzado por las explosiones, de este modo, descubriremos nuevos mundos y lugares extraños que recorrer.

3.5. CONCEPT ART

LOS ROBOTS

En este caso, he creado un robot futurista. La idea es que las articulaciones y la cabeza floten en el aire, como si estuviese compuesto por imanes. Además, le he dado un aspecto amigable con unos guantes de boxeo y unas zapatillas de deporte.



23 - Boceto del personaje principal. Creado en Illustrator

LOS ESCENARIOS

Las escenas de huida se componen de un camino infinito sobre el que se van creando diferentes obstáculos. Este camino estará rodeado por una ambientación diferente en cada nivel. Para ello, hemos utilizado elementos gratuitos de la Asset Store de Unity.

Por ejemplo, hemos utilizado el Pack Sci-Fi Style Modular Pack para los laboratorios de fabricación de robots y parte del escenario.

A medida que avanzamos de nivel, vamos descubriendo nuevos mundos con Assets totalmente diferentes y con nuevas ambientaciones.



24 - Imagen demo de Sci-fi Style Modular Pack.

De Unity Asset Store. [Online]<<https://assetstore.unity.com/packages/3d/environments/sci-fi/sci-fi-styled-modular-pack-82913>>

3.6. OBJETIVOS Y MECÁNICAS

El objetivo del juego es llegar lo más lejos posible evitando una serie de obstáculos que irán apareciendo de forma aleatoria. De este modo, obtendremos puntos que nos colocarán en lo más alto de la clasificación.

MOVIMIENTOS:

El personaje se desplazará sobre tres carriles y tendrá que esquivar saltando o rodando una serie de obstáculos hasta llegar a una plataforma segura y superar el nivel.

NIVELES:

El juego constará de 3 niveles. Los dos primeros, serán a modo tutorial y, para desbloquear cada uno de ellos, tendremos que superar el nivel anterior.

El tercer nivel será infinito. En este nivel, nuestro personaje morirá tarde o temprano y, la misión del jugador, será alejarlo todo lo posible de la explosión y obtener la máxima puntuación.

PUNTOS:

En el tercer nivel del juego, por cada metro que avancemos conseguiremos puntos. Estos puntos se almacenan en el dispositivo dónde estamos ejecutando el juego. Por lo que en el menú principal podremos ver cuál es la mejor puntuación y así, intentar superarla.

TIEMPO

En los primeros niveles, cada vez que choquemos con un obstáculo, perderemos algo de tiempo, si nos quedamos sin tiempo, el robot será destruido y no desbloquearemos el siguiente nivel.

VIDA

Existe en la parte superior derecha una barra que muestra la vida que le queda al robot. Si esa barra desaparece, significará que el robot ha perdido su vida y será destruido.

Cada vez que choquemos con un obstáculo perderemos parte de esta vida.

DIFICULTAD

En el menú principal podremos elegir entre tres niveles de dificultad.

Esto hará que varíe la velocidad del juego y la visualización de los obstáculos (Cuanto más difícil, más curvo es el escenario y, por consiguiente, menor la visibilidad).

VELOCIDAD

En todos los niveles y dependiendo de la dificultad, la velocidad se va incrementando para hacer más difícil el juego progresivamente.

3.7. DESARROLLO - UNITY 3D

Al tratarse de un juego en 3D, los primeros motores en los que he pensado han sido Unity 3D y Unreal Engine. Ambos motores disponen de una versión gratuita siempre que no superemos unos determinados ingresos.

Personalmente, estoy más familiarizado con Unity 3D y con el lenguaje de programación C#, por lo que este ha sido el motor elegido para utilizar en la realización de este juego.

Este motor permite crear de forma rápida todos los elementos necesarios para la realización de la aplicación:

- Animaciones
- Físicas
- Interacción
- Iluminación

- Scripting
- Audio
- Video
- UI
- Etc.

Unity dispone de una gran cantidad de assets para utilizar en nuestros proyectos de forma gratuita, así como una gran comunidad de desarrollo en la que podemos obtener infinidad de tutoriales y ejemplos de todo tipo.

Además, nos permite exportar el resultado a un gran número de plataformas.

3.8. ROADMAP

1 - CREACIÓN DEL PERSONAJE PRINCIPAL

Idea y bocetado del modelo a diseñar.

Modelado en Blender.

Asignación de materiales y texturas.

2 - ANIMACIÓN

Exportaremos el personaje desde Blender para importarlo en Mixamo, de este modo crearemos un “Rigging” de forma rápida y podremos exportar una gran cantidad de animaciones de todo tipo para utilizar en el juego.

Seleccionamos las animaciones básicas.

MECÁNICA E INTERACCIÓN

Crearemos una escena base con la mecánica del juego. Definiremos la posición de la cámara, los movimientos del jugador y la forma de avanzar. También configuraremos el modo en el que el jugador interactúa con el juego.

4 - MENÚ

Creamos el menú principal y las distintas opciones. También crearemos el menú de selección de niveles.

5 - DESARROLLO DEL NIVEL 1

Sobre la escena base crearemos el primer nivel funcional del juego.

6 - EVALUACIÓN CON USUARIOS

Haré pruebas con usuarios expertos para identificar posibles mejoras o cambios.

7 - ALMACENAMIENTO DE RESULTADOS.

Configuraremos la forma en la que el juego guarda la puntuación y almacena los resultados en el equipo.

8 - INTRODUCCIÓN

Crearemos la escena de introducción del juego mediante animaciones y Assets.

9 - DESARROLLO DEL NIVEL 2

Crearemos el segundo nivel del juego.

10 - DESARROLLO DEL NIVEL 3

Crearemos el tercer nivel del juego.

11 - EVALUACIÓN CON USUARIOS

Volvemos a probar el juego con diferentes usuarios menos expertos que nos aporten un punto de vista diferente sobre posibles mejoras.

12 - RETOQUES FINALES.

Añadiremos detalles decorativos, animaciones o cualquier otra cosa que de un valor añadido al juego.

4. DISEÑO TÉCNICO

4.1. ENTORNO DE DESARROLLO – UNITY 2019

El entorno de trabajo escogido para el desarrollo de este videojuego ha sido Unity.

Unity es un motor gráfico tanto 3D como 2D con un entorno de fácil manejo y con un gran potencial. Su utilización es gratuita con algunas restricciones. También cuenta con una versión de pago.

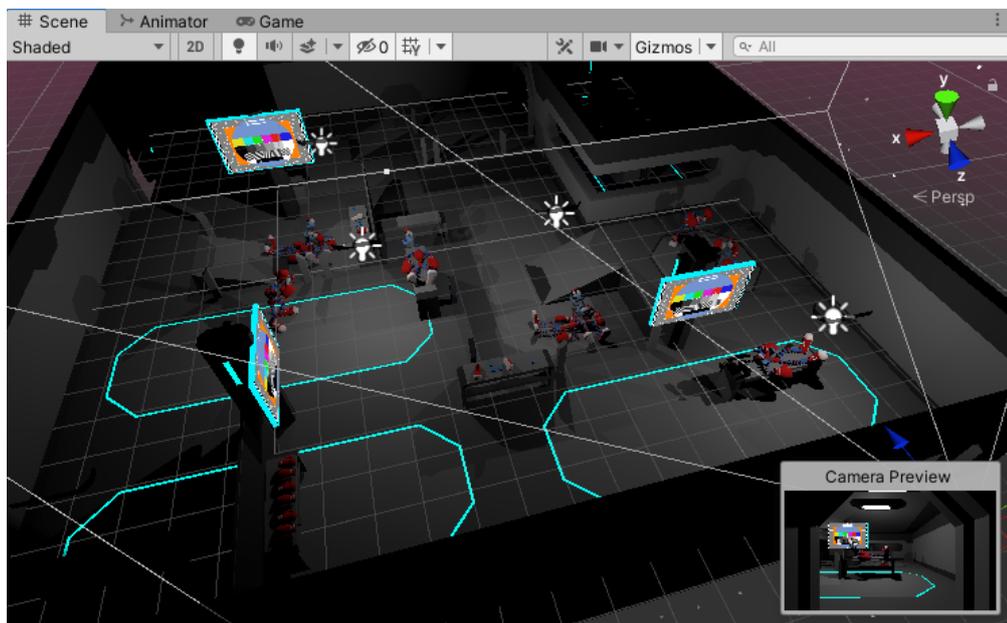
Es una herramienta multiplataforma, lo que nos permitirá compilar nuestro videojuego para múltiples sistemas operativos y dispositivos.

PANELES

[9] El entorno de trabajo se divide en paneles que podemos modificar a nuestro gusto. Existen diversos paneles, los más importantes o utilizados serían:

SCENE

Es el espacio 3D/2D dónde situamos y componemos la escena con los objetos que tenemos en nuestro proyecto. En mi caso, en este panel es dónde he montado los diferentes niveles y la introducción:



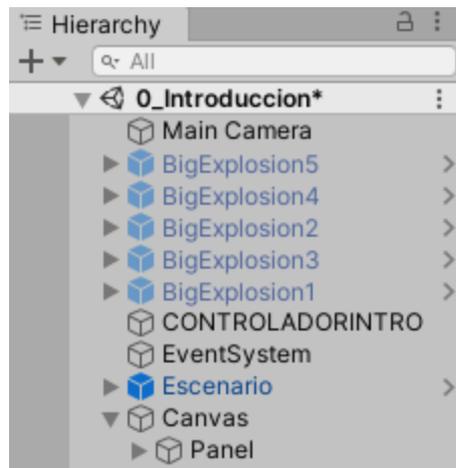
25 – Panel Scene de Unity.

Imagen capturada desde el propio software.

HIERARCHY

Listado de objetos que tenemos dentro de nuestra escena. Aquí tenemos una organización de todos los activos de la escena. Muchos de estos objetos, pueden contener otros como hijos suyos de este modo, se le aplicarán comportamientos similares a los del objeto padre. En este caso, yo he utilizado también objetos

vacíos para organizar esta vista. Por ejemplo, dentro del objeto vacío “Escenario”, he incluido todos los objetos que conforman el escenario dentro de la escena.



26 – Panel Hierarchy de Unity.

Imagen capturada desde el propio software.

PROJECT

Listado de todos los activos que podemos incorporar a nuestra escena. Aquí podemos observar todos los assets que podemos utilizar en el proyecto.

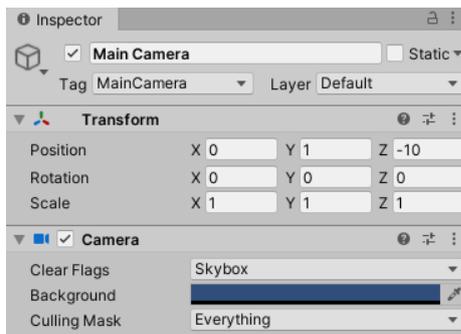


27 – Panel Project de Unity.

Imagen capturada desde el propio software.

INSPECTOR

Muestra las propiedades de los objetos. Aquí añadimos los scripts, las animaciones o cualquier otra propiedad que queremos que tenga el objeto.

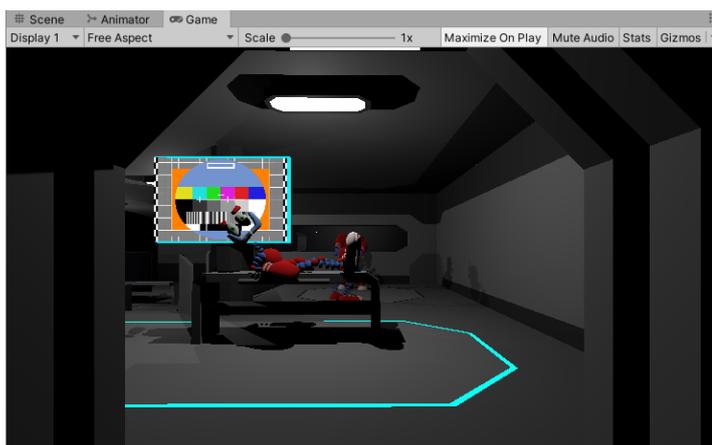


28 – Panel Inspector de Unity.

Imagen capturada desde el propio software.

GAME

Vista de la cámara activa del juego cuando lo ejecutamos. Muestra lo que verá el jugador.

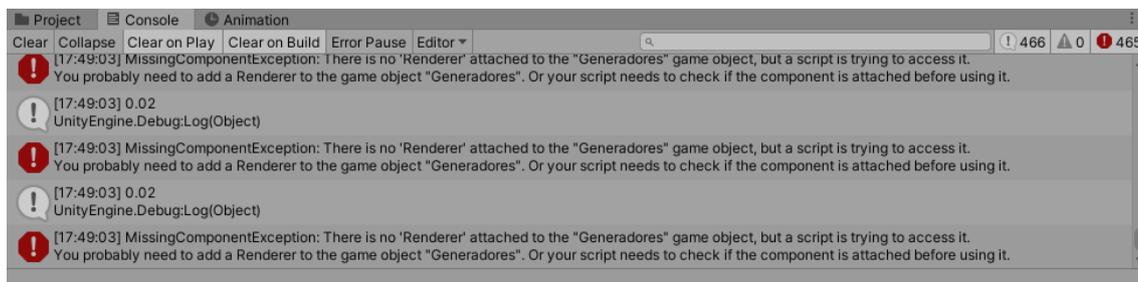


29 - Panel Game de Unity.

Imagen capturada desde el propio software.

CONSOLE

Muestra mensajes y errores del proyecto.



30 - Panel Console de Unity.

Imagen capturada desde el propio software.

OTROS

Existen otros paneles que iremos cargando conforme los vayamos utilizando como “Animation”, “Animator”, “Timeline”, “Audio Mixer”, “Asset Store”, etc.

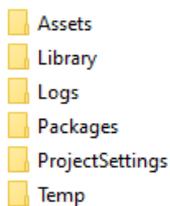


31 - Otros paneles de Unity.

Imagen capturada desde el propio software.

PROYECTOS

Cuando creamos un nuevo proyecto, Unity genera una carpeta en la que almacenará todos los activos que utilizaremos en el desarrollo del videojuego.



ASSETS

[10] Entendemos como Asset, cualquier activo que utilizamos en nuestro juego. Estos activos pueden ser modelos3D, Sprites, Texturas, escenas, Sonidos, fuentes, scripts, etc.

Unity acepta todos los formatos de archivo de los programas más populares que existen en la actualidad como Blender, Maya o 3Ds Max. Así como archivos de sonido .wav, .mp3 o .aif e imágenes en PNG, JPEG o TIFF.

El resto de las carpetas contienen metadatos, ficheros temporales, paquetes y logs que utiliza Unity para su correcto funcionamiento.

Dentro de estos Assets, aparte de todos los modelos 3D o Sprites que podemos introducir, Unity nos permite utilizar algunos propios entre los que podemos destacar los siguientes:

ESCENAS

Las escenas de Unity son el lugar dónde creamos los diferentes niveles de nuestro juego y dónde colocaremos nuestros activos dentro del espacio 3D o 2D. En este caso, tenemos 6 escenas:

1. 0_Introducción
2. 1_MenuPrincipal
3. 2_MenuNiveles
4. 3_Nivel1
5. 4_Nivel2
6. 5_Nivel3

También podemos considerar las escenas como activos de Unity, por lo que las almacenamos también en la carpeta Assets del proyecto.

SCRIPS

Los scripts son programas simples que se encargan de dotar de vida a nuestros activos. Podemos asignar varios scripts a cada objeto y desde aquí es desde dónde programaremos su comportamiento.

Los lenguajes de programación utilizados para programar los scripts en Unity son Javascript y C#.



32 – Scripts. Imagen capturada desde el propio software.

```

ControllIntro.cs -> X
Assembly-CSharp
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5  using UnityEngine.UI;
6
7  public class ControlIntro : MonoBehaviour
8  {
9      public GameObject explosion1;
10     public GameObject explosion2;
11     public GameObject explosion3;
12     public GameObject explosion4;
13     public GameObject explosion5;
14     public GameObject robot1;
15     public GameObject robot2;
16     public GameObject robot3;
17     public AudioSource Explosion;
18
19     Animator anim1;
20     Animator anim2;
21     Animator anim3;
22
23     // Start is called before the first frame update
24
25     private void Awake()
26     {
27         Screen.sleepTimeout = SleepTimeout.NeverSleep;
28     }

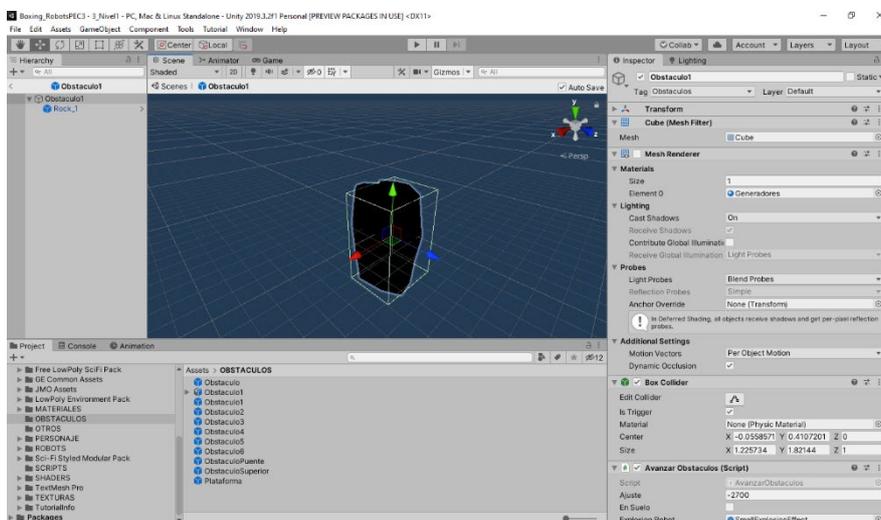
```

33 – Código C# de Unity.
Imagen capturada desde el propio software.

PREFABS

Cuando modificamos las propiedades de activo o le asignamos unos determinados componentes. Podemos almacenar estos nuevos activos en nuestra carpeta de Assets de forma que podamos volver a utilizarlos más tarde con esa misma configuración. A estos activos los denominamos Prefabs.

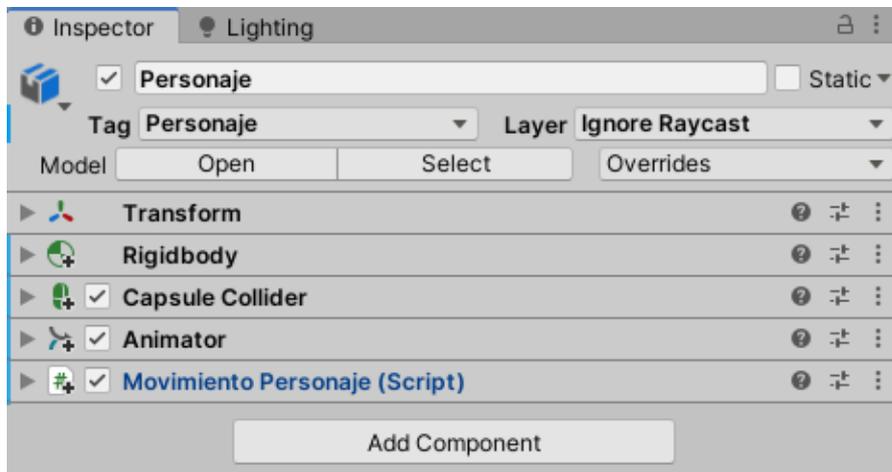
Por ejemplo, en nuestro proyecto, hemos utilizado los prefabs para crear los obstáculos que se van generando automáticamente durante el nivel.



34 – Prefabs de Unity.
Imagen capturada desde el propio software.

COMPONENTES

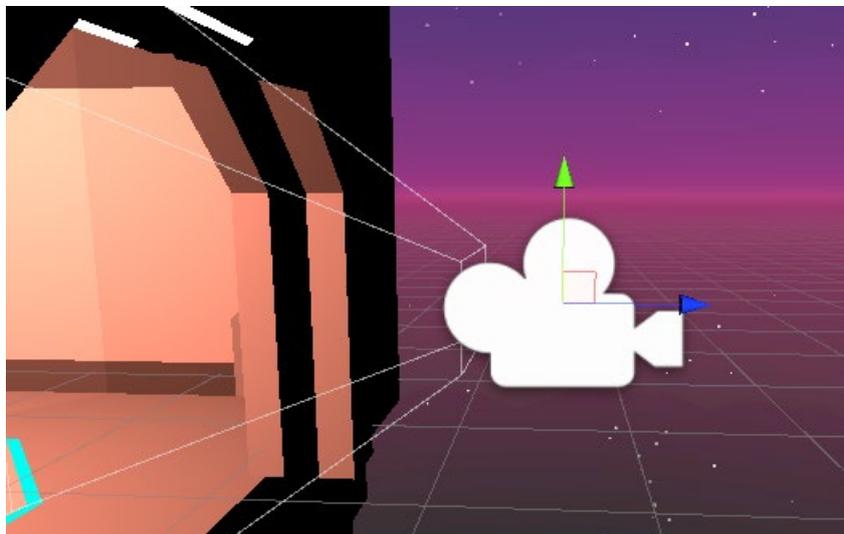
Con los componentes añadimos o modificamos las propiedades de los objetos que introducimos en las escenas. También podemos añadir nuevos componentes a un objeto para añadirle funcionalidades, estos pueden ser animaciones, scripts, etc.



35 - Componentes de Unity.
Imagen capturada desde el propio software.

CÁMARAS

Las cámaras muestran al jugador la información del juego para permitirle interactuar con él. En cada escena podremos crear múltiples cámaras y renderizar parte o la totalidad del contenido de una escena por pantalla.



36 - Cámara de Unity.
Imagen capturada desde el propio software.

4.2. REQUISITOS TÉCNICOS

UNITY EDITOR

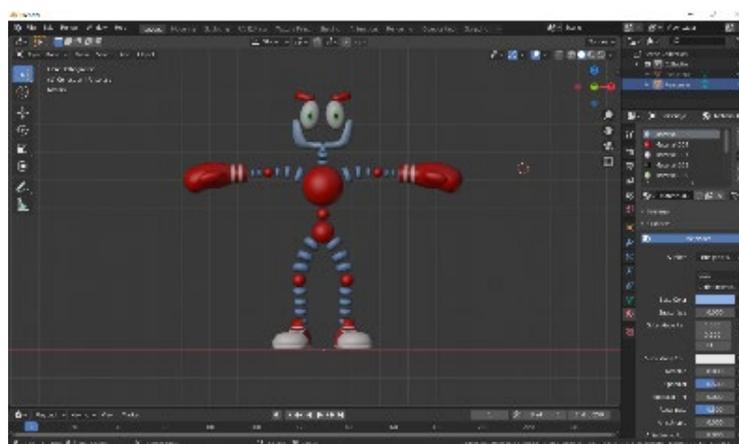
[11] Los requisitos mínimos para utilizar el editor de Unity son los siguientes:

Plataforma	SO	CPU	API Gráficos	Otros
Windows	Windows (SP1+)	7 Arquitectura x64 con SSE2	DX10, DX11, DX12 – GPUs capaces.	Controladores oficiales soportados.
	Windows x64	10		
MacOS	Sierra 10.12.6+	Arquitectura x64 con SSE2	Metal – Intel y AMD GPUs capaces.	Controladores oficiales soportados.
Linux	Ubuntu 16.04	Arquitectura x64	OpenGL 3.2+	Controlador Oficial Nvidia o AMD Mesa
	Ubuntu 18.04			
	CentOS7	Con SSE2	Intel y AMD GPUs capaces.	

4.3. HERRAMIENTAS EMPLEADAS.

BLENDER

El personaje principal del videojuego ha sido modelado en Blender 2.82a

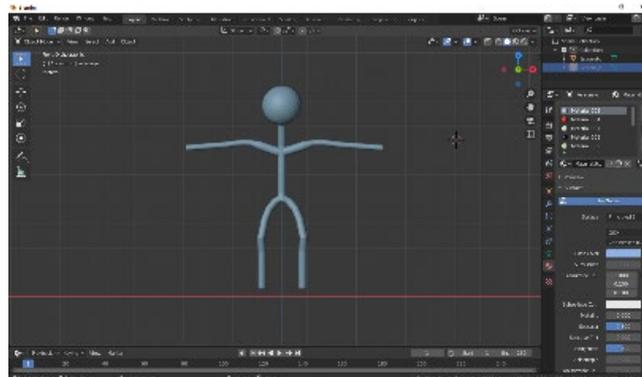


37 - Modelo 3D del Robot en Blender.
Imagen capturada desde el propio software.

Blender es un software libre que permite la creación de gráficos tridimensionales. Así como la transformación de estos, su animación, renderizado o su iluminación entre otras cosas.

Para la creación de este he utilizado el modificador “mirror”, y así, he podido crear el personaje más fácilmente de una forma totalmente simétrica en T-pose. Esto ayudará posteriormente a aplicarle un esqueleto.

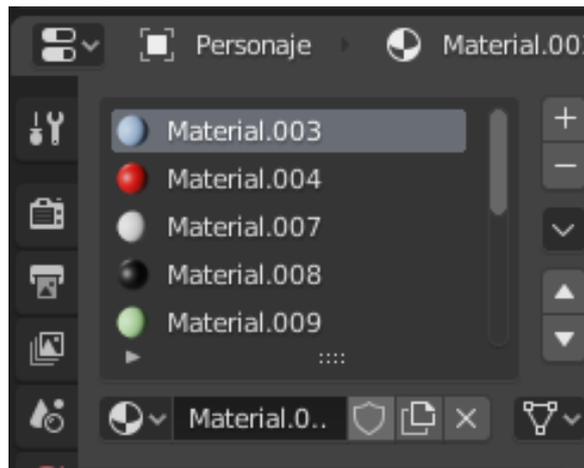
He añadido un “esqueleto” para unir todas las partes, ya que, de otro modo, Mixamo no importaba el modelo correctamente al estar separadas las piezas entre sí.



38 - Esqueleto del robot para Mixamo.

Imagen capturada desde el propio software.

También he utilizado Blender para asignarle diversos materiales al personaje. En este caso, no he utilizado texturas.



39 - Materiales asignados al robot en Blender.

Imagen capturada desde el propio software.

Por último, he exportado el personaje en formato FBX.

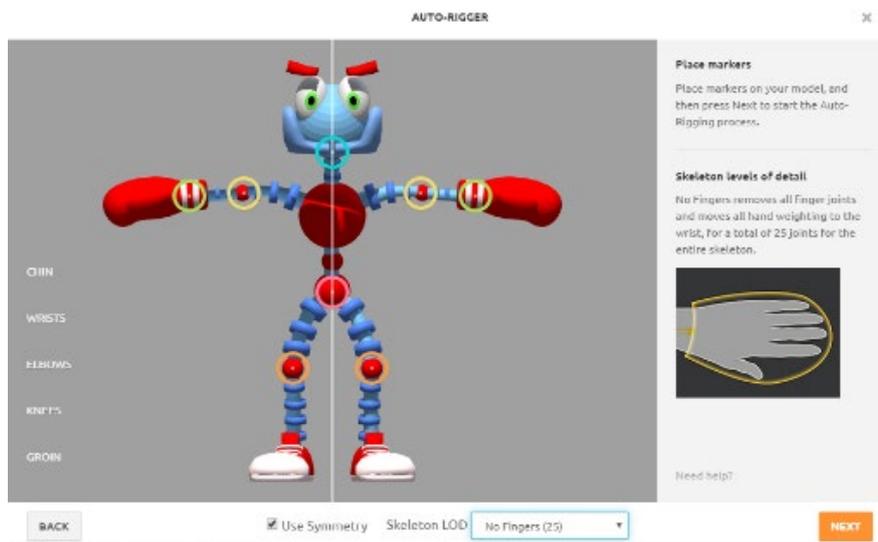


40 - Resultado del modelado 3D del robot en Blender.
Imagen capturada desde el propio software.

MIXAMO

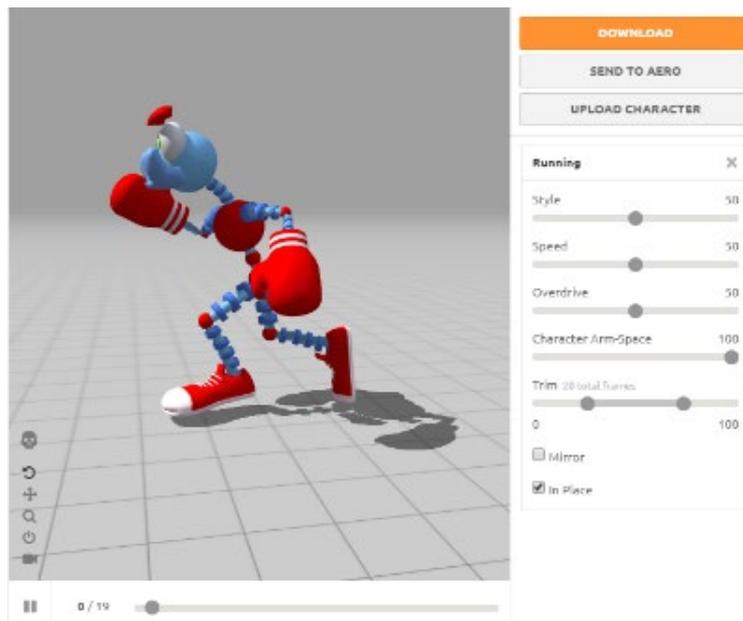
Para la animación del personaje. Importamos desde la web de Mixamo (<https://www.mixamo.com/>) el fichero FBX generado en Blender. Marcamos los puntos clave para la colocación de los huesos y, como por arte de magia, Mixamo creará el esqueleto en cuestión de segundos.

Al crear al personaje simétricamente en Blender, encontraremos menos problemas en este proceso.



41 - Puntos clave de la animación en Mixamo.
Imagen capturada desde el propio software.

A continuación, buscamos la animación “running”, la seleccionamos y la aplicamos al personaje.



42 - Animación "Running" de Mixamo.
Imagen capturada desde el propio software.

Una vez ajustados los parámetros, podemos descargar el personaje seleccionando las opciones “.FBX for Unity(.fbx)” y “Whit Skin” para cargarlo en Unity.

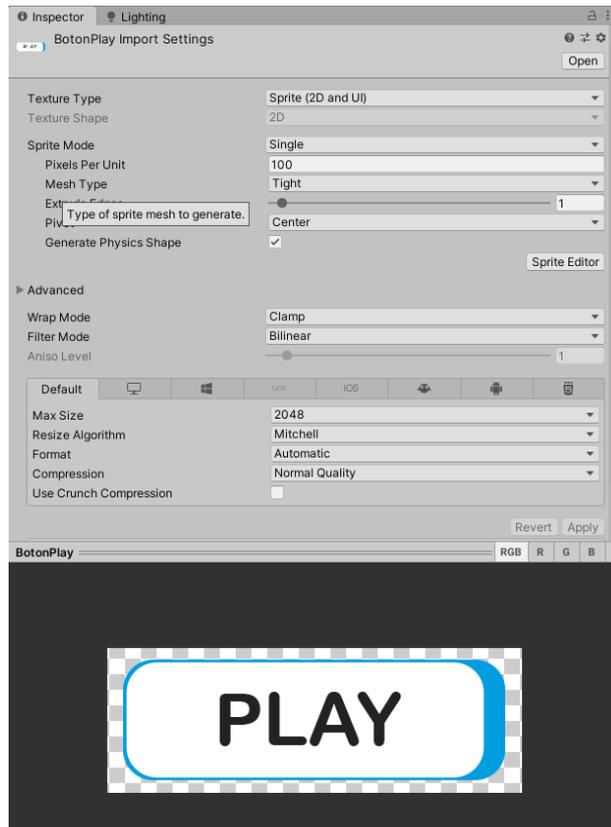
El resto de las animaciones del personaje las iremos descargando sin el skin, puesto que ya tenemos el modelo en Unity.

PHOTOSHOP

El diseño de los botones e imágenes 2D utilizadas en el juego, se han creado mediante las herramientas Photoshop. Estas imágenes las he exportado a PNG manteniendo el canal Alfa. Una vez en Unity, he importado los archivos y los he configurado como Sprites (2D and UI) para poder utilizarlos dentro del “Canvas”.



43 - Botón Play en Photoshop.
Imagen capturada desde el propio software.

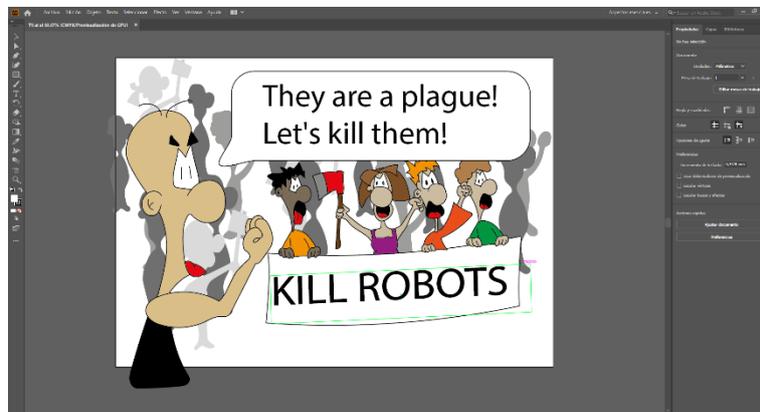


44 - Botón Play en Unity.

Imagen capturada desde el propio software.

ILLUSTRATOR

En *Illustrator* he creado las imágenes vectoriales para el vídeo que aparece en las pantallas de la introducción. He utilizado una tableta Wacom Intuos 4 para dibujar los personajes a mano. Después los he retocado ligeramente para ajustar los trazos y colorear las figuras.



45 - imagen para pantallas TV de la introducción en Illustrator.

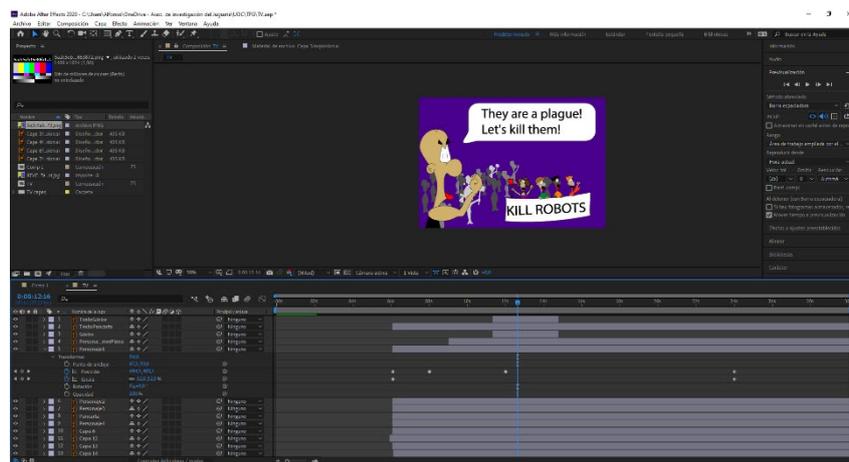
Imagen capturada desde el propio software.

AFTER EFFECTS

He utilizado *After Effects* para crear la animación que parece en las pantallas de la introducción. La finalidad de este vídeo es situar al jugador en contexto y mostrar la situación de los robots y cómo los humanos se manifiestan contra ellos como si de las noticias se tratase.

Para ello, he cargado el fichero de *Illustrator* creado anteriormente directamente para que crear todas las capas. Después, he animado la posición de los personajes para hacerles avanzar por la calle. He añadido al personaje principal para potenciar más las reivindicaciones de los humanos.

También he añadido explosiones y efectos de distorsión para aportar dramatismo a la escena.



46 - Animación en After Effects.

Imagen capturada desde el propio software.



47 - Vista de la animación en Unity.

Imagen capturada desde el propio software.

4.4. INVENTARIO DE RECURSOS

RECURSOS EXTERNOS:

- **FREE Skybox Extended Shader:** Lo he utilizado para incorporar un skybox a las escenas de los niveles 1 y 2.
Obtenido de la Asset Store:
<https://assetstore.unity.com/packages/vfx/shaders/free-skybox-extended-shader-107400>
- **Fantasy Skybox FREE:** Utilizado para incorporar el skybox del 3er nivel.
Obtenido de la Asset Store:
<https://assetstore.unity.com/packages/2d/textures-materials/sky/fantasy-skybox-free-18353>
- **Unity Particle Pack:** Lo he utilizado para incorporar las explosiones en el proyecto.
Obtenido de la Asset Store:
<https://assetstore.unity.com/packages/essentials/tutorial-projects/unity-particle-pack-127325>
- **Sci-Fi Styled Modula Pack:** Lo he utilizado para crear la meta de los niveles.
Obtenido de la Asset Store:
<https://assetstore.unity.com/packages/3d/environments/sci-fi/sci-fi-styled-modular-pack-82913>
- **Free LowPoly Scifi Pack:** Lo he utilizado para la creación del laboratorio en la introducción.
Obtenido de la Asset Store:
<https://assetstore.unity.com/packages/3d/environments/sci-fi/free-lowpoly-scifi-110070>
- **Curved World:** Shader que he utilizado para curvar el escenario.
Obtenido en internet de:
<http://blog.onebyonedesign.com/games/unity3d-endless-runner-part-i-curved-worlds/>
- **Imagen calle fondo** (De uso libre): La he utilizado para el fondo de la animación de After Effects, para las pantallas de TV de la introducción.
Obtenido de internet de:
<https://www.freepng.es/png-cn1wwa/download.html>
- **Stylized Planet-Pack Full:** Lo he utilizado para crear los planetas que orbitan alrededor en los niveles 1 y 2.
Obtenido de la Asset Store:
<https://assetstore.unity.com/packages/3d/environments/stylized-planet-pack-full-148233>

- **LowPoly Environment Pack:** Utilizado para agregar los objetos del escenario del 3er nivel.
Obtenido de la Asset Store:
<https://assetstore.unity.com/packages/3d/environments/landscapes/lowpoly-environment-pack-99479>
- **Los sonidos** al abrir el panel opciones, al producirse una explosión fuerte, al chocar con los obstáculos, al rodar, al saltar o al moverse, al derrumbarse el laboratorio o el checkpoint, cuando quedas eliminado o cuando ganas, cuando hay fuego, o al seleccionar o pulsar cualquier botón.
Estos sonidos los he agregado en el juego para que suenen cuando realizamos alguna acción o sucede algo en el juego.
Obtenidos de internet:
<http://sonidosmp3gratis.com/>
- **Música Introducción:** Música de fondo de la introducción.
Obtenida de Youtube Audio Library:
<https://www.youtube.com/audiolibrary/music?nv=1>
- **Música de fondo:** Música de fondo de los niveles.
Obtenida de Youtube Audio Library:
<https://www.youtube.com/audiolibrary/music?nv=1>

RECURSOS PROPIOS

- **Animación 2D:** Vídeo de animación 2D para las TVs de la introducción.

La animación de las Televisiones que aparecen en la introducción la he desarrollado en After Effects. Para ello, he creado los gráficos vectoriales en Illustrator y los he separado en capas.

He importado el archivo de Illustrator en After Effects y he animado cada una de las capas para obtener un vídeo de animación 2D.
- **Robot 3D:** Modelo 3D del robot.

El robot lo he modelado íntegramente en Blender a partir de un boceto inventado. He intentado que sea de baja poligonización para que el juego sea bastante fluido y le he dado forma humana para poder incorporarle animaciones posteriormente de forma fácil.

Los materiales se los he asignado también en Blender.

También he utilizado partes del robot para decorar el escenario de la introducción.

- **Botones: Saltar Intro, Play, Options, Atrás, Niveles, Locked.**

Los he creado mediante Illustrator y Photoshop. Exportándolos finalmente a PNG (En caso de tener transparencias) o JPG (Sin transparencias) para usarlos en la aplicación.

- **Imágenes: Título, Fondo del panel opciones, Puntuación máxima,**

Los he creado mediante Illustrator y Photoshop. Exportándolos finalmente a PNG (En caso de tener transparencias) o JPG (Sin transparencias) para usarlos en la aplicación.

4.5. ARQUITECTURA DEL JUEGO

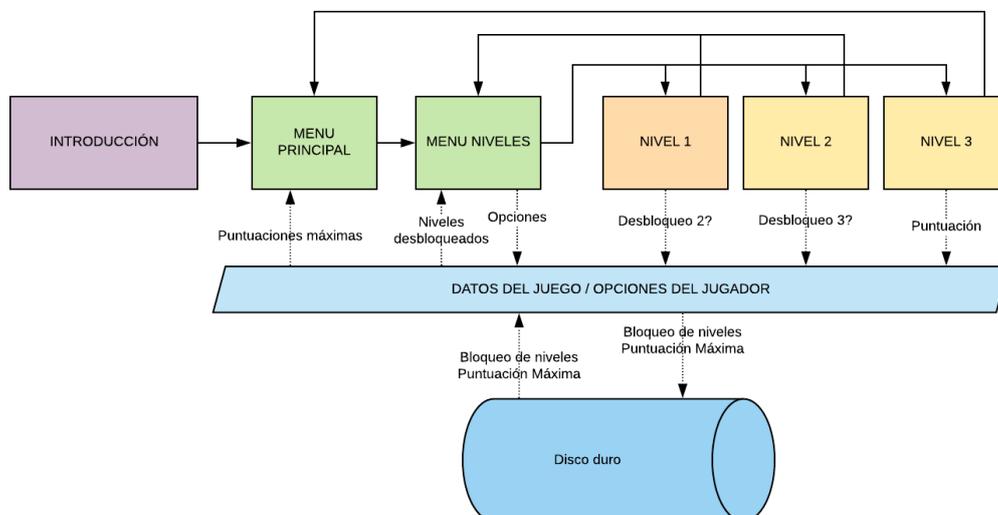
El juego comenzará con una introducción que dará paso al menú principal. Aquí, cargaremos los datos desde el disco duro para obtener las puntuaciones máximas.

A continuación, almacenaremos las variables con las opciones del jugador.

En el menú de niveles, cargaremos los datos para ver que niveles están bloqueados y cuáles no.

Una vez elegimos un nivel, pasamos a jugar el nivel.

Al finalizar este, guardaremos si hemos superado el nivel o no y, en el caso del tercer nivel, guardaremos la puntuación.



48 - Arquitectura del videojuego.

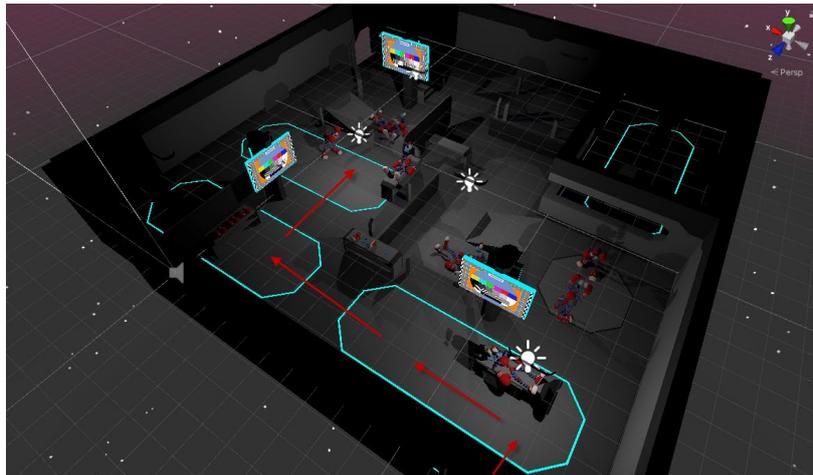
Creado en Lucidchart.

5. DISEÑO DE NIVELES

5.1. PRIMERA ESCENA – LA INTRODUCCIÓN

Aunque no se trate de un nivel con interacción en sí por parte del jugador, la introducción se ha diseñado del mismo modo creando un escenario 3D y moviéndonos por su interior cómo si fuésemos uno de los personajes del juego.

Inicialmente, la cámara se mueve por el escenario observando lo que hay a su alrededor:

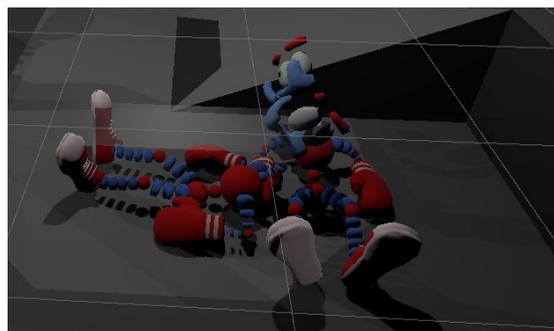


49 - El escenario de la introducción.
Captura desde el editor de Unity.

En el escenario se han añadido varias pantallas para mostrar la situación que se está viviendo en ese momento. También se han incluido piezas de los robots, robots entrenando y robots destrozados o listos para ser reparados.



50 - Guantes de robot.
Captura desde el editor de Unity.



51 - Robots destrozados.
Captura desde el editor de Unity.

Con este diseño se pretende introducir al jugador en un ambiente de tensión e intranquilidad. La luz ténue y el ambiente belico ayudarán a aumentar la tensión del jugador.

Del mismo modo, el vivir la escena en primera persona hará que el jugador se sumerja mucho más en la experiencia.

Finalmente, el punto álgido de la introducción será el momento en el que comienzan las explosiones y el jugador huye rápidamente de las instalaciones. El aumento de la velocidad de los movimientos, los sonidos y las explosiones aumentan rápidamente la tensión preparando al jugador para iniciar la partida.

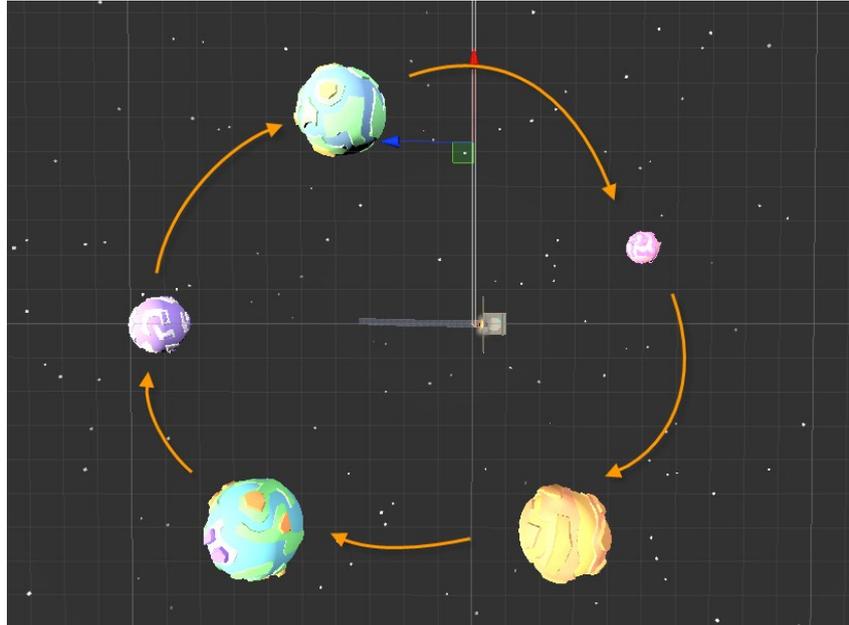
5.2. NIVEL 1 – SALIDA DEL LABORATORIO

En el primer nivel, el robot aparece en la puerta del laboratorio del que ha huido y vemos como todo se desmorona tras él.



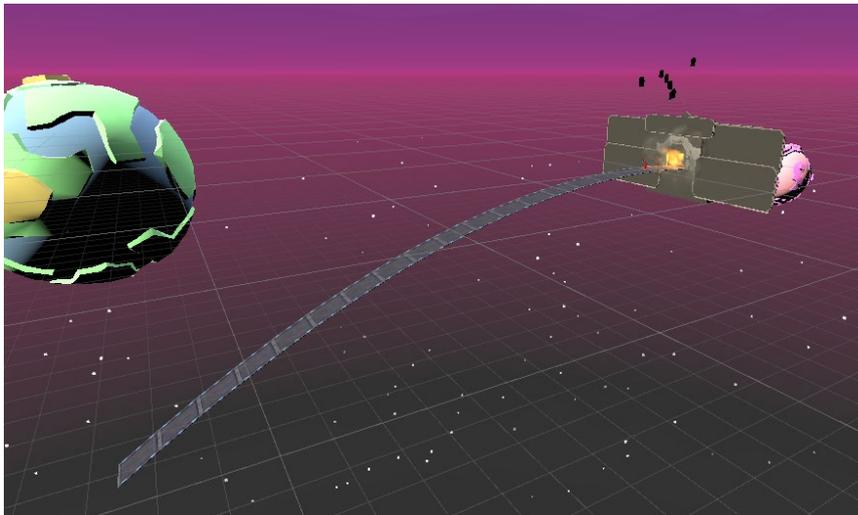
52 - Destrucción del laboratorio.
Captura de pantalla del juego.

El escenario está compuesto por un camino a través del espacio por el que avanzamos constantemente evitando chocar con rocas que caen procedentes de la explosión. Además, aparecen una serie de planetas orbitando alrededor del personaje para ambientar la escena.



53 - Planetas Orbitando.
Captura del editor de Unity.

Como añadido, el camino que recorreremos va curvándose en diferentes direcciones para dar una perspectiva diferente al jugador en cada momento. Durante este nivel, el jugador únicamente necesitará mover al personaje de izquierda a derecha y saltar.

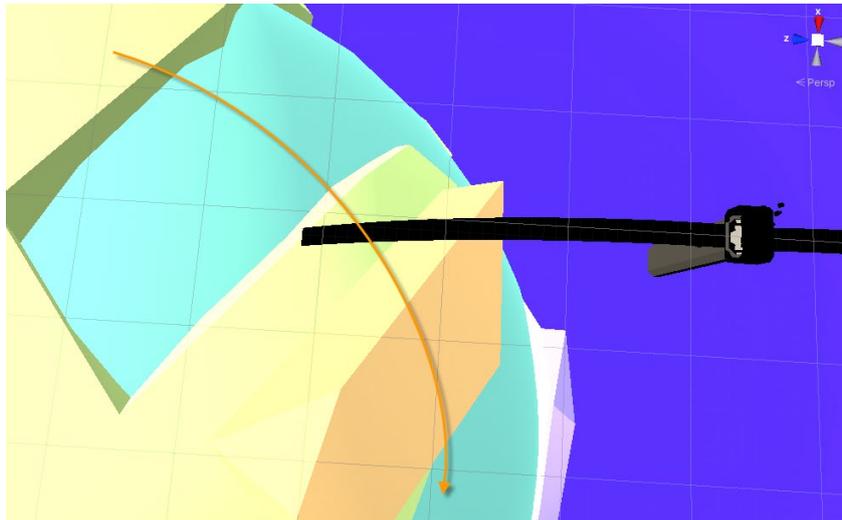


54 - Camino curvado.
Captura desde el editor de Unity.

5.3. NIVEL 2 – DESCENDIENDO A LA TIERRA

En el segundo nivel, el personaje comienza desde la meta del nivel anterior. La cual, queda destruida inmediatamente al empezar.

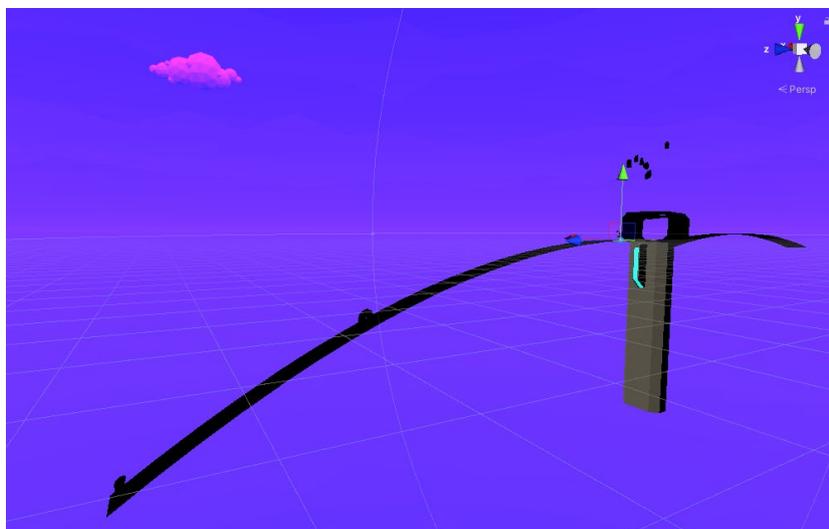
En este nivel cambia el cielo, puesto que nos adentramos en la atmósfera terrestre. La tierra se ve mucho más cerca y podemos apreciar su rotación.



55 - Rotación de la tierra.

Captura desde el editor de Unity.

Además, en este nivel se han introducido nuevos obstáculos para que el jugador ahora también tenga que rodar. El camino va curvándose al igual que en el nivel anterior.



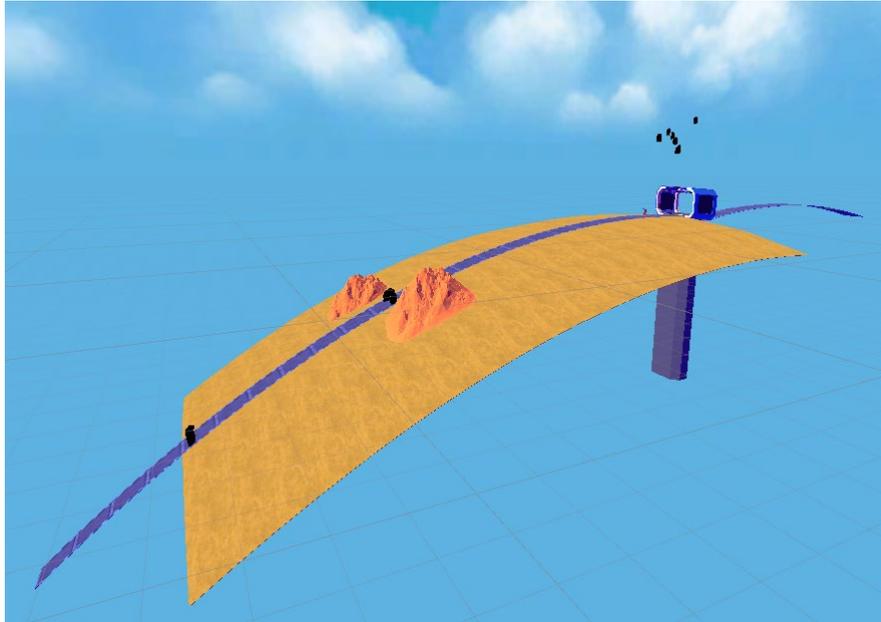
56 - Nivel 2.

Captura desde el editor de Unity.

5.4. NIVEL 3 – EN LA TIERRA

En este último nivel, al igual que en el anterior, el personaje aparece frente a la meta del nivel anterior que, igualmente, es destruida al instante.

El decorado se va generando automática y aleatoriamente frente al jugador. Aparecen montañas, árboles y rocas. Del mismo modo que en los niveles anteriores, la curva del escenario irá variando continuamente. El cielo también cambia, ya estamos en la tierra.



57 - Nivel 3.

Captura desde el editor de Unity.

Este nivel es infinito y, tarde o temprano, el jugador será eliminado ya que, la velocidad se va incrementando hasta hacer imposible que el personaje esquive los obstáculos.

6. MANUAL DE USUARIO

6.1. REQUISITOS TÉCNICOS PARA JUGAR

A TRAVÉS DE UNITY PLAYER PARA PC

Respecto a Unity Player, los requisitos mínimos para la versión de escritorio que necesitamos para ejecutar el proyecto los enumeramos a continuación.

Estos requerimientos variarán en función de la complejidad del proyecto. En este caso, sería suficiente cumplir con los siguientes:

Plataforma	SO	CPU	API Gráficos	Otros
Windows	Windows (SP1+)	7 Arquitectura x86 o x64 con SSE2	DX10, DX11, DX12 – GPUs capaces.	Controladores oficiales soportados.
	Windows x64	10		

Para poder jugar, tendremos que descargar los archivos del juego desde el siguiente enlace:

<https://github.com/Alfonsoibiuoc/BoxingRobots/tree/master/BoxingRobotsGame/Ejecutable%20PC>

Una vez descargados, únicamente tendremos que ejecutar el archivo “BoxingRobots.exe”

DISPOSITIVOS ANDROID

[12] En este caso, la aplicación se ha compilado para dispositivos móviles con una versión de Android igual o superior a la 4.4 “Kit Kat” (API level 19).

En este caso, se utilizan las APIs OpenGL ES3 y Vulkan para el renderizado, las cuales sólo son compatibles con Android a partir de la versión 4.3.6.2. Instrucciones

Para poder jugar, tendremos que descargar el archivo apk del siguiente enlace:

<https://github.com/Alfonsoibiuoc/BoxingRobots/tree/master/BoxingRobotsGame/APK%20Android>

Para instalar el archivo, tendremos que habilitar en el dispositivo la opción de “Orígenes desconocidos” y posteriormente ejecutar el archivo descargado.

7. CONCLUSIONES

El desarrollo de un producto multimedia, como es el caso del videojuego que he desarrollado, abarca una variedad de disciplinas técnicas muy amplia.

Para la creación del personaje, se han utilizado técnicas de modelado 3D, de rigging, de animación e incluso de creación de materiales y texturas. Se han programado sus movimientos teniendo en cuenta la física, se ha programado la interacción de los botones, las colisiones de los objetos y todas las animaciones de los menús. Se han serializado datos para almacenarlos en el dispositivo. Se han utilizado técnicas de diseño de interfaces y de diseño gráfico para la realización de los botones y elementos 2D que aparecen a lo largo de la aplicación. Se ha editado vídeo y audio, se ha generado un corto de animación 2D y, todo eso, sin olvidarnos de una correcta gestión del proyecto y de la correcta integración de todos los contenidos mediante Unity.

Si intentamos realizar todo esto “sobre la marcha”, lo más probable es que no podamos finalizar correctamente el proyecto o, incluso, que podamos llegar a no finalizarlo nunca. Es por ello, que es necesaria una correcta planificación del trabajo desde el principio.

Creo que este es el concepto clave para la realización de un proyecto de esta envergadura, la planificación.

Una vez tenemos la idea y la descomponemos en partes para marcar las diferentes fases, es esencial marcar unas metas, un tiempo y unos recursos para llevar a cabo el desarrollo de esta. Este es otro aspecto importante que he tenido muy en cuenta durante la realización del trabajo, la organización.

Estructurar bien todos los elementos y las relaciones que hay entre ellos nos ayudará a corregir rápidamente posibles errores. Nos permitirá dimensionar la aplicación en cualquier momento o realizar modificaciones sin volvernos locos.

Como lección, diría que la planificación y la organización son primordiales y, casi suponen la mitad del trabajo.

En el caso del videojuego realizado, creo que esta planificación ha sido buena y se han podido cumplir los objetivos a tiempo.

El objetivo principal era la realización de un videojuego que planteara una historia original y, creo que, en este sentido, he conseguido abordarlo correctamente. Además, se han alcanzado otras metas como la realización de un videojuego corto pero adictivo, funcional y, visualmente atractivo.

Un objetivo que ha cambiado ligeramente desde el inicio de la aplicación es la plataforma de destino. Inicialmente, el videojuego únicamente estaría disponible para PC. Sin embargo, finalmente pude modificar los controles y adaptarlos a plataformas Android rápidamente.

Creo que este era un punto a favor que aportaría un valor añadido al proyecto sin un coste de tiempo demasiado alto.

8. GLOSARIO

ACELERMETRO

Se denomina acelerómetro a cualquier instrumento destinado a medir aceleraciones.

Wikipedia: Acelerómetro [En línea]

<<https://es.wikipedia.org/wiki/Aceler%C3%B3metro>> [06/05/2020].

APK

Los APK son archivos ejecutables para Android. Beben del formato .jar de Java, y permiten instalar componentes empaquetados en el sistema.

Xataka: Qué es un APK de Android, cómo se instala y diferencias con las aplicaciones normales. [En línea] <<https://www.xatakandroid.com/aplicaciones-android/que-apk-android-como-se-instala-diferencias-apps-normales>> [06/05/2020]

C SHARP

Lenguaje de programación multiparadigma desarrollado y estandarizado por Microsoft como parte de su plataforma .NET.

Wikipedia: C Sharp [En línea] <https://es.wikipedia.org/wiki/C_Sharp> [06/05/2020].

CANAL ALFA

En computación gráfica, la composición alfa o canal alfa es la que define la opacidad de un píxel en una imagen.

Wikipedia: Composición alfa [En línea]

<https://es.wikipedia.org/wiki/Composici%C3%B3n_alfa> [06/05/2020]

CANVAS DE UNITY

El Canvas es el área donde todos los elementos UI deben estar. El Canvas es un Game Object con un componente Canvas en él, y todos los elementos UI deben ser hijos de dicho Canvas.

Unity Documentation: Canvas [En línea]

<<https://docs.unity3d.com/es/2018.4/Manual/UICanvas.html>> [06/05/2020]

COMPILAR

tr. Inform. Convertir un programa en lenguaje máquina a partir de otro programa de computadora escrito en otro lenguaje.

REAL ACADEMIA ESPAÑOLA: DICCIONARIO DE LA LENGUA ESPAÑOLA, 23.ª ed., [versión 23.3 en línea]. <<https://dle.rae.es>> [06/05/2020].

GAME ENGINE

El término motor de videojuego (en inglés *game engine*), o simplemente motor de juego, hace referencia a una serie de rutinas de programación que permiten el diseño, la creación y el funcionamiento de un videojuego.

Wikipedia: Motor de videojuego [En línea]
<https://es.wikipedia.org/wiki/Motor_de_videojuego> [06/05/2020].

GIROSCOPIO

El giróscopo o giroscopio es un dispositivo mecánico que sirve para medir, mantener o cambiar la orientación en el espacio de algún aparato o vehículo.

Wikipedia: Giróscopo [En línea] <<https://es.wikipedia.org/wiki/Gir%C3%B3scopo>> [06/05/2020].

GRÁFICO VECTORIAL

Una imagen vectorial es una imagen digital formada por objetos geométricos dependientes (segmentos, polígonos, arcos, muros, etc.), cada uno de ellos definido por atributos matemáticos de forma, de posición, etc.

Wikipedia: Gráfico vectorial [En línea]
<https://es.wikipedia.org/wiki/Gr%C3%A1fico_vectorial> [06/05/2020]

HTML5

Quinta revisión importante del lenguaje básico de la World Wide Web, HTML.

Wikipedia: HTML5 [En línea] <<https://es.wikipedia.org/wiki/HTML5>> [06/05/2020].

JAVASCRIPT

Lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Wikipedia: Javascript [En línea]
<<https://es.wikipedia.org/wiki/JavaScript>> [06/05/2020].

LOG

En informática, se usa el término registro, *log* o historial de *log* para referirse a la grabación secuencial en un archivo o en una base de datos de todos los acontecimientos (eventos o acciones) que afectan a un proceso particular (aplicación, actividad de una red informática, etc.). De esta forma constituye una evidencia del comportamiento del sistema.

Wikipedia: Log (informática) [En línea]
<[https://es.wikipedia.org/wiki/Log_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Log_(inform%C3%A1tica))> [06/05/2020]

MULTIMEDIA

adj. Que utiliza conjunta y simultáneamente diversos medios, como imágenes, sonidos y texto, en la transmisión de una información.

REAL ACADEMIA ESPAÑOLA: DICCIONARIO DE LA LENGUA ESPAÑOLA, 23.ª ed., [versión 23.3 en línea]. <<https://dle.rae.es>> [06/05/2020].

MULTIPLATAFORMA

En informática, multiplataforma es un atributo conferido a programas informáticos o métodos y conceptos de cómputo que son implementados, y operan internamente en múltiples plataformas informáticas.

Wikipedia: Multiplataforma [En línea] <<https://es.wikipedia.org/wiki/Multiplataforma>> [06/05/2020].

PLUGIN

En informática, un complemento o «plug-in» es una aplicación (o programa informático) que se relaciona con otra para agregarle una función nueva y generalmente muy específica.

Wikipedia: Complemento (informática) [En línea] <[https://es.wikipedia.org/wiki/Complemento_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Complemento_(inform%C3%A1tica))> [06/05/2020].

RENDERIZACIÓN

Proceso de generar una imagen fotorrealista o no fotorrealista a partir de un modelo 2D o 3D (o modelos en lo que colectivamente podría llamarse un archivo de escena) por medio de programas informáticos.

Wikipedia: Renderización [En línea] <<https://es.wikipedia.org/wiki/Renderizaci%C3%B3n>> [06/05/2020].

RIGGING

El rigging es el proceso de crear un sistema de controles digitales y agregárselos a un modelo 3D para que así pueda ser animado fácil y eficientemente.

Wikipedia: Rigging [En línea] <<https://es.wikipedia.org/wiki/Rigging>> [06/05/2020]

SERIALIZAR

Proceso de codificación de un objeto en un medio de almacenamiento.

Wikipedia: Serialización [En línea] <<https://es.wikipedia.org/wiki/Seriazizaci%C3%B3n>> [06/05/2020]

SPRITE

Tipo de mapa de bits dibujados en la pantalla de ordenador por hardware gráfico especializado sin cálculos adicionales de la CPU.

Wikipedia: Sprite (Videojuegos) [En línea] <[https://es.wikipedia.org/wiki/Sprite_\(videojuegos\)](https://es.wikipedia.org/wiki/Sprite_(videojuegos))> [06/05/2020]

9. BIBLIOGRAFÍA

- [1]. **GamerDic: Endless Runner.** – GamerDic
<<http://www.gamerdic.es/termino/endless-runner/>>
- [2]. Agudo, Sergio: **¿Qué son los juegos Endless Runner: ¡A correr y correr!** - Malavida. <<https://www.malavida.com/es/analisis/que-son-los-juegos-endless-runner-a-correr-y-correr-006670#gref>>
- [3]. **Diagrama de Gantt de seguimiento de fechas** – Microsoft
<<https://templates.office.com/es-es/diagrama-de-gantt-de-seguimiento-de-fechas-tm22588720>>
- [4]. **Yoyo Games – Game Maker Studio 2**
<<https://www.yoyogames.com/gamemaker/features>>
- [5]. **Construct3 - Game Making Software** <<https://www.construct.net/en>>
- [6]. **Unity – Crea un mundo con más juego.**
- [7]. **Unreal Engine – Make something Unreal**
<<https://www.unrealengine.com/en-US/>>
- [8]. **Unity forum – Swipe in all directions Touch and Mouse**
<<https://forum.unity.com/threads/swipe-in-all-directions-touch-and-mouse.165416/>>
- [9]. **Unity Documentation – Aprendiendo la interfaz**
<<https://docs.unity3d.com/es/2018.4/Manual/LearningtheInterface.html>>
- [10]. **Tokio School – Tipos de Assets en Unity3D**
<https://www.tokioschool.com/noticias/tipos-assets-unity3d/>
- [11]. **Unity Documentation – System requirements for Unity 2019.3**
<<https://docs.unity3d.com/Manual/system-requirements.html>>
- [12]. **Android Developer – Open GL ES**
<https://developer.android.com/guide/topics/graphics/opengl>