



THROUGH TIME

Abel González Ramos

*Grado en Técnicas de Interacción Digital i Multimedia
20.466 - TFG Videojuegos*

Consultor: JOEL SERVITJA FEU

Profesor responsable de la asignatura: JOAN ARNEDO MORENO

Enero 2022



Esta obra está sujeta a una licencia de [Reconocimiento-NoComercial-SinObraDerivada 3.0 España](https://creativecommons.org/licenses/by-nc-nd/3.0/es/) de Creative Commons

FICHA DEL TRABAJO FINAL

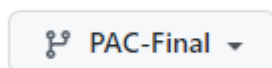
Título del trabajo:	<i>Through time</i>
Nombre del autor:	<i>Abel González Ramos</i>
Nombre del consultor/a:	<i>Joel Servitja Feu</i>
Nombre del PRA:	<i>Joan Arnedo Moreno</i>
Fecha de entrega (mm/aaaa):	<i>01/2022</i>
Titulación o programa:	<i>Grado en Técnicas de Interacción Digital i Multimedia</i>
Área del Trabajo Final:	<i>20.466 - TFG Videojuegos</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Unity, videojuego, plataformas, interactivo</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación metodología, resultados i conclusiones del trabajo.</i>	
<p>El objetivo principal del TFG ha sido crear Through time, un videojuego indie de plataformas 2D en un entorno 3D, donde el personaje tiene que realizar una serie de interacciones con diferentes elementos en los que se puede actuar en el nivel para llegar a su objetivo final y en el que durante su aventura tendrá de evitar a los enemigos que le puedan atacar u obstaculizar el paso. También se puede conseguir como extra, todos los ítems coleccionables “Lorguins” que se encuentran en las zonas más ocultas del nivel.</p> <p>Se ha utilizado el motor gráfico de Unity con el lenguaje de programación C#. El videojuego está preparado para poderlo jugar con teclado y gamepad, con lo que se puede implementar para cualquier plataforma que permita estos periféricos como PC o consolas como PS4 o XBOX, entre otros.</p> <p>En conclusión, se han llegado a los objetivos principales marcados, aunque nos ha faltado que el personaje pudiera disparar a los enemigos, pero por temas de tiempo no ha sido posible hacerlo. Por lo demás, estoy satisfecho con el trabajo realizado pudiendo crear un juego desde cero dibujando y modelando todos los diseños, dándole un toque interactivo con los elementos del escenario y la introducción de una historia de fondo interesante de viajes temporales.</p>	

Abstract (in English, 250 words or less):

The main objective of the TFG has been to create Through time, an indie 2D platform videogame in a 3D environment, where the character has to perform a number of interactions with different elements in which he can act in the level to reach his final goal and in which during the adventure, he will have to avoid the enemies that can attack him or hinder the passing. You can also get as extra, all the collectible items "Lorguins" found in the most hidden areas of the level.

The Unity graphics engine has been used with the C# programming language. The videogame is prepared to be played with a keyboard and gamepad, so it can be implemented for any platform that allows these peripherals such as PC or consoles like PS4 or XBOX, among others.

To sum up, the target set has been reached, although we have lacked that the character could shoot the enemies, as on account of time issues it hasn't been possible to do it so. Otherwise, I am really satisfied with the work done being able to create a game from scratch by drawing and modeling all the designs, giving it an interactive touch with the elements of the scene, and the introduction of an interesting backstory of temporary trips.

Entregables:**ENLACE AL REPOSITORIO GITHUB:**

<https://github.com/GonRam/tfg-uoc.git>

Importante: En el repositorio, para descargar la última versión del videojuego, se tiene de ir al **branche PAC-Final**, allí esta comprimido por partes el ejecutable listo para ser descargado.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo	1
1.2 Objetivos del Trabajo.....	1
1.3 Enfoque y método seguido	1
1.4 Planificación del trabajo.....	2
1.5 Breve resumen de productos obtenidos	3
1.6 Breve descripción de los otros capítulos de la memoria.....	3
2. Estado de arte.....	4
2.1 Género del juego	4
2.2 Plataformas de desarrollo.....	4
3. Definición del juego.....	6
3.1 Descripción del juego	6
3.2 Historia, ambientación i/o trama	6
3.3 Definición de los personajes/ elementos.....	6
3.4 Interacción entre los actores del juego.....	7
3.5 Objetos planteados al jugador	7
3.6 Concept art	8
4. Diseño técnico.....	9
4.1 Entorno desarrollo elegido	9
4.2 Requisitos técnicos del entorno de desarrollo.....	9
4.3 Inventario y descripción de las herramientas utilizadas.....	9
4.4 Inventario y descripción de los assets y recursos al juego.....	10
4.5 Esquema de arquitectura del juego/ componentes.....	15
4.6 Funcionamiento de la IA de los enemigos.....	15
5. Diseño de niveles.....	16
5.1 Apartamentos Tower5.....	16
5.2. Plaza central.....	16
5.3 Torre mirador.....	17



5.4 Central eléctrica.....	17
6. Análisis de costes	18
7. Manual de usuario	19
7.1 Requerimientos mínimos técnicos del sistema.....	19
7.2 Instrucciones del juego.....	19
8. Conclusiones.....	20
9. Glosario	21
10. Bibliografía.....	22

Lista de Figuras

<i>Ilustración 1. Roadmap del proyecto</i>	2
<i>Ilustración 2. Diagrama de Gantt del proyecto</i>	2
<i>Ilustración 3. Disney's Hercules - PSX (1997)</i>	4
<i>Ilustración 4. Tintín en el templo del sol MS-DOS (1997)</i>	4
<i>Ilustración 5. Beneath a Steel Sky – MS-DOS (1994)</i>	4
<i>Ilustración 6. Flashback - MS-DOS (1992)</i>	4
<i>Ilustración 7. Imágenes del videojuego Beyond a Steel Sky</i>	8
<i>Ilustración 8. Bocetos de algunos edificios</i>	8
<i>Ilustración 9. John Baker – Personaje principal</i>	10
<i>Ilustración 10. John Baker – Personaje principal</i>	10
<i>Ilustración 11. Diseño de personajes secundarios</i>	11
<i>Ilustración 12. Diseño de enemigos</i>	11
<i>Ilustración 13. Diseño ítems en 3D y de la UI</i>	11
<i>Ilustración 14. UI Objeto interactivo</i>	12
<i>Ilustración 15. Fechas indicación movimiento jugador</i>	12
<i>Ilustración 16. Diseño anuncios y soporte publicitario</i>	12
<i>Ilustración 17. Efecto humo i círculo fuego</i>	12
<i>Ilustración 18. Diseño fondos escena del nivel</i>	13
<i>Ilustración 19. Diseño edificios del escenario</i>	13
<i>Ilustración 20. Diseño elementos del escenario</i>	13
<i>Ilustración 21. Diseño pantallas opciones</i>	13
<i>Ilustración 22. Pantallas de los videos del juego</i>	14
<i>Ilustración 23. Diagrama de flujo del juego</i>	15
<i>Ilustración 24. Diseño zona apartamentos</i>	16
<i>Ilustración 25. Diseño zona plaza central</i>	16
<i>Ilustración 26. Diseño zona Torre mirador</i>	16
<i>Ilustración 27. Diseño zona Central eléctrica</i>	17
<i>Ilustración 28. Controles del juego</i>	19

1. Introducción

1.1 Contexto y Justificación del Trabajo

El mundo de los videojuegos ha evolucionado en su historia, tanto visualmente como a nivel de jugabilidad ya que las consolas u ordenadores de los 80 y 90 al tener limitaciones en su hardware hacían que los juegos fueran mucho más cortos y algunos terriblemente complicados de completarlos para alargar su vida útil. Entre juegos de esos años podríamos destacar las primeras entregas de Rayman, Metal Slug, Crash Bandicoot, entre muchos otros.

Trouth Time se encuentra entre esos juegos de plataformas que mezclan elementos 2D con 3D, en el que se ha intentado cuidar el apartado visual y darle una historia de fondo con un video introductorio para atrapar al jugador en el juego. Además, tiene una parte interactiva en la que permite hablar o utilizar objetos del escenario y que si no se realiza una acción previa no se puede realizar la siguiente y así sucesivamente hasta llegar al final del nivel.

1.2 Objetivos del Trabajo

Los objetivos planteados en el trabajo han sido los siguientes:

- Creación de un menú principal 3D con las opciones de jugar, créditos, controles, ajustes o salir.
- Diseño y programación de la pantalla de opciones, pausa, Game Over o Fin de juego dentro del nivel del videojuego.
- Diseño del video introductorio, de los créditos y de las acciones que realiza el personaje como pulsar un botón, abrir una rejilla, etc.
- Diseño y programación de las interacciones del personaje principal, como los controles de movimiento y acciones interactiva.
- Creación y diseño del nivel del juego con las localizaciones principales donde tendrá de ir el personaje (Apartamentos, Torre mirado, Central eléctrica, plaza central y parque).
- Creación de dos tipos de enemigos con dos tipos de comportamiento y sus animaciones. Uno que al detectar al jugador haga daño a este y continúe su patrulla y otro con una IA básica que cuando vea al jugador lo ataque hasta eliminarlo o se aleje este del rango de la ruta que está patrullando.
- Diseño de diálogos de ayuda e indicaciones de los objetivos al jugador.
- Creación de obstáculos como una trampa laser en la central eléctrica.

1.3 Enfoque y método seguido

En la creación de un videojuego se requiere el diseño de la parte gráfica, modelado y la de programación y testeo de este. Aunque el tiempo nos ha limitado de realizar un producto más completo y con mas funcionalidades, hemos optado por realizar un videojuego nuevo basándonos en muchos diseños de un juego de aventura gráfica ya existente, pero rediseñándolos desde cero con Illustrator y con una historia totalmente diferente.

Se ha usado música y efectos sonoros obtenidos de recursos gratuitos y las voces se han gravado con micrófono para después ser editados con Audition.

El método que se ha seguido para el desarrollo a sido el siguiente:

Primero se ha planteado el tipo de juego, la historia de este, el estilo de diseño que se le quiere dar y los objetivos principales y secundarios que tiene de hacer el jugador para completar el nivel. También se ha hecho un estudio de que motor gráfico utilizar, eligiendo al final Unity por su fácil aprendizaje y por el gran nombre de tutoriales que se encuentran de ayuda.

Por último, se han creado y recopilado recursos, diseñado el nivel y animaciones, creado todos los scripts y se han realizado las pruebas y correcciones para al final ya tener un el producto terminado.

1.4 Planificación del trabajo

Las tareas se han planificado en base a las entregas a realizar en la asignatura.

- **PAC 1:** En esta PAC se ha desarrollado la idea y conceptualización la historia, personajes, mecánicas y objetivos del juego.
- **PAC 2:** Se ha diseñado los personajes, el menú principal y paneles UI del juego y el propio nivel programando las mecánicas y las funcionalidades básicas.
- **PAC 3:** Se sigue añadiendo y mejorando el nivel corrigiendo errores detectados y dando nuevas funcionalidades al personaje. Añadimos animaciones al escenario, elementos interactivos y enemigos con IA Básica.
- **PAC Final:** Se corrigen errores y se realizan algunas mejoras, se aplican sonidos a los interactivos y realizan últimos detalles visuales.

Realizamos Roadmap indicando las tareas a realizar y se adjunta también el diagrama de Gantt del proyecto.



Ilustración 1. Roadmap del proyecto

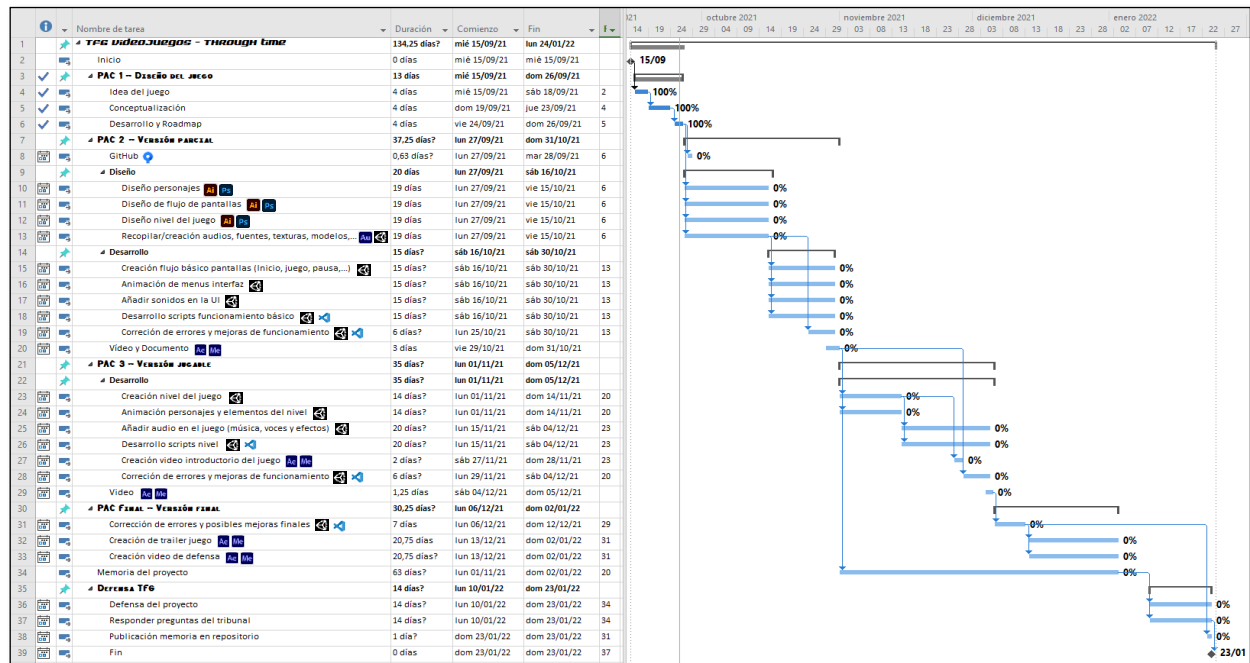


Ilustración 2. Diagrama de Gantt del proyecto

1.5 Breve resumen de productos obtenidos

Los productos obtenidos a lo largo del curso han sido:

- Versión final ejecutable del videojuego.
- Videos explicativos de la PAC 2 y PAC 3.
- Archivos de desarrollo del trabajo (scripts, sprites, prefabs, sonidos, fuentes, etc.).
- Video de defensa del proyecto.
- Informe de autoevaluación.
- Memoria del proyecto.

1.6 Breve descripción de los otros capítulos de la memoria

- **Capítulo 2:** Estado del arte

Se explica la evolución del género de nuestro videojuego y los motores gráficos actuales.

- **Capítulo 3:** Definición del juego

Se explica conceptualmente nuestro videojuego, centrándonos en los objetivos, mecánicas y características de los personajes y la historia.

- **Capítulo 4:** Diseño técnico

Aquí se explica la parte más técnica indicando las herramientas utilizadas, etc.

- **Capítulo 5:** Diseño de niveles

Se muestra e mapa del nivel y explica los criterios de diseño.

- **Capítulo 6:** Análisis de costos

Se indican los costes del trabajo y los recursos energéticos consumidos.

- **Capítulo 7:** Manual de usuario

Se muestran los requisitos técnicos e instrucciones del videojuego.

- **Capítulo 8:** Conclusiones

Conclusiones finales del proyecto.

- **Capítulo 9:** Glosario

Definiciones de la terminología utilizada en el proyecto.

- **Capítulo 10:** Bibliografía/Referencias

Recursos bibliográficos utilizados en el desarrollo del documento.

2. Estado de arte

2.1 Genero del juego

El subgénero en el que me he basado del videojuego es el de plataformas, tenemos en su historia juegos clásicos de la década de los 80 y 90 como *Donkey Kong* y *Super Mario Bros*.

El jugador puede moverse en diferentes profundidades para evitar obstáculos, un ejemplo sería el juego de *Tintín en el templo del sol* en el que en muchos niveles Tintín tiene de hacer estas acciones para evitar enemigos u obstáculos para llegar a sus objetivos.

El juego de *Disney's Hercules* nos ayudó a la idea de combinar el 2D con los fondos 3D utilizando sprites en nuestro juego, ya que me permite realizar fácilmente los personajes y enemigos y poderlos animar para el movimiento o el dialogo.

En la ambientación me he basado a juegos como *Flashback* y por otro lado al juego de aventura grafica *Beneath a Steel Sky* con el que se pueden realizar algunas acciones interactivas como hablar con personajes o accediendo a nuevas zonas según si se ha cogido o realizado ciertas acciones. La parte de las cinemáticas que hemos realizado al pulsar botones nos hemos cogido de referencia *Flashback*.



Ilustración 3. *Tintín en el templo del sol* MS-DOS (1997)



Ilustración 4. *Disney's Hercules* - PSX (1997)



Ilustración 5. *Beneath a Steel Sky* – MS-DOS (1994)

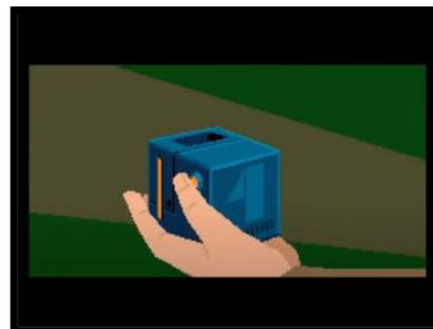


Ilustración 6. *Flashback* - MS-DOS (1992)

2.2 Plataformas de desarrollo

Hay muchos engines o middlewares para el desarrollo de juegos 3D y 2D, así que lo primero que se ha hecho es un estudio de cuál es el más adecuado para nuestro proyecto y que la curva de aprendizaje sea demasiado compleja por la limitación de tiempo.



Cocos2D es un framework escrito en Python para crear juegos 2D, presentaciones gráficas y recientemente ya incluye un motor 3D con Cocos Editor 3.0. Desde 2018 existe para varias plataformas o lenguajes como *Cocos2D-XNA* para soporte XNA en C# o *Cocos2d-Js* para soporte de JavaScript entre muchos otros. Es de código abierto y gratuito, con lo que es muy usado. Entre otros juegos que utilizan este framework tenemos Ghost Archer o Run, giant!



GAMEMAKER es una plataforma basada en un lenguaje de programación interpretado y un kit de desarrollo de software (SDK) para desarrollar videojuegos. Utiliza un lenguaje de programación de scripts llamado Game Maker Language (GML) i tiene una versión gratuita, pero de tan solo 30 días. Entre otros juegos que utilizan este sistema tenemos Hotline Miami, Hero Siege o Valdis Story.



UNREAL es un motor creado por Epic Games y utilizado en sus inicios en el shooter en primera persona Unreal en el año 1998. Desde el 2015 se puede descargar gratuitamente, aunque si el proyecto creado recauda más de 3.000 dólares por trimestre la empresa pedirá un 5% de sus ganancias. Es uno de los motores más potentes i utilizados, entre otros juegos que utilizan este motor son Shenmue 3 o Gears of war 4. También se ha de decir que una vez realizado el proyecto te permite la posibilidad de exportarlo a diferentes plataformas como Nintendo Switch, PS5, Xbox One, PC y dispositivos móviles en Android e IOS.



UNITY es uno de los motores 2D i 3D más conocidos y utilizados, además tiene mucha documentación disponible ya sea en su [página oficial](#) [7] como videotutorials a Youtube, llibres, etc. Tiene su versión gratuita con la que se tiene de cumplir como requerimiento que no se gane mas de 100.000 dólares al año. La programación de los scripts se puede hacer con el lenguaje C# o javascript. Entre otros juegos que utilizan este motor son Pokémon Go, Ori and the Blind Forest, Superhot, Cuphead o Rust. Como con Unreal, también se tiene la posibilidad de exportar el proyecto en Nintendo Switch, PS5, Xbox One, PC y dispositivos móviles en Android e IOS.

3. Definición del juego

3.1 Descripción del juego

Es un juego de plataformas con un entorno 3D y se han utilizado sprites para crear los personajes muchos elementos del juego. El personaje principal aunque va lateralmente en algunos lugares pasar a un nivel de profundidad más cercano o lejano para investigar, evitar ciertos enemigos u objetos que le van apareciendo a su paso. Por otro lado, puede realizar algunas interacciones sencillas con el escenario como hablar con otros personajes para obtener cierta información para pasar a otra zona o accionar ciertos objetos para activar por ejemplo el funcionamiento de un ascensor.

3.2 Historia, ambientación I/o trama

La ambientación se encuentra dentro del género **ciberpunk**, una película precursora de este género fue Blade Runner del 1982 dirigida por Ridley Scott. Por otro lado, la trama introduce los viajes temporales, aunque en el proyecto el tiempo disponible solo nos permitirá realizar un primer nivel en el futuro, dejando la historia abierta a un sinfín de posibilidades en la historia y en los diversos niveles que se puede llegar a crear.

La historia empieza en el futuro, en la ciudad de Orion City en el año 2222, donde un policía novato llamado John Baker que está en ese momento en su apartamento recibe una llamada de un desconocido que se dice llamar Ryan y el cual según dice fue amigo de su padre. Le indica que tiene de contarle algo que podría afectar a la historia y a toda la humanidad. El juego termina antes que pueda terminar de contarle que esta ocurriendo y la pregunta es puede fiar de él.

3.3 Definición de los personajes/ elementos

Personajes:

John Baker: Es el personaje jugable principal de la historia. Lleva un uniforme de policía y una mochila en la que podrá ir guardando objetos que se vaya encontrado para poder utilizar mas tarde. Si el tiempo lo permite también tendrá una pistola para poder disparar a los enemigos.

Ryan: Es un personaje que conocemos solo empezar la historia, aunque solo tenemos un contacto telefónico y no lo veremos al menos en esta demo del juego. Este ayudara a John durante la historia, aunque John no sabe si fiarse del todo de él, de momento le va haciendo caso ya que sabe mucha cosa y conforme pasando la historia descubre que son ciertas. Aunque sabe que también le está ocultando cierta información.

Enemigos: John tendrá de evitar dos tipos de enemigos, los robots voladores que tienen una IA básica y un robot de limpieza que simplemente esquivándolo evitaremos que nos haga daño.

Elementos:

En el escenario abra elementos fijos o en movimiento obstaculizando el paso al jugador como chispas eléctricas, caídas al vacío, zonas resbaladizas, objetos que caen de la fachada o en movimiento.



Objeto interactivo: Hay ciertos objetos que podemos interactuar pulsando el botón ver o utilizar/ hablar.



Créditos: John en el nivel encontrara estas monedas que le servirán para comprar objetos.



Lorguin: Son coleccionables que se encuentran repartidos por todo el nivel, hay un máximo de diez en total.



Vida: En algunos puntos del escenario hay ítems que permitirá al jugador recuperar un porcentaje de vida perdida.

3.4 Interacción entre los actores del juego



El personaje las interacciones que puede realizar en los objetos es con un botón para mirar este objeto/personaje y este se pone a hablar describiendo el objeto. Con otro el botón actuar si este objeto lo permite realizará una acción y lo indicará con un texto en pantalla, audio hablando el personaje y una animación realizando dicha acción. También con este mismo botón se puede hablar con personajes, en el

videojuego podremos hablar con Ryan.

Si se encuentra a enemigos o las trampas laser el jugador las tendrá de esquivar ya que sino John ira perdiendo vida en la barra y si llega a cero saltará el fin de juego y se tendrá de volver a empezar desde cero.

3.5 Objetos planteados al jugador

Indicamos los objetivos que tiene de hacer el jugador para llegar al final del nivel:

Objetivo PRINCIPAL: El jugador tiene el objetivo principal de que la vida no se le quede a 0% al moverse por el escenario ya que si se muere tendrá de volver a empezar de nuevo.

Objetivo 1. Buscar ordenador público para hablar con Ryan.

Hay varios ordenadores repartidos por el nivel, el jugador tendrá de buscar cual es el correcto. Uno de ellos, el de la central da una pista de donde hay un ordenador inaccesible en la azotea de la Torre Mirador. Al pulsar el sensor del ascensor no hará nada.

Objetivo 2. Buscar la forma de activar la corriente del ascensor.

John tendrá de ir a la Central eléctrica y entrar por una rejilla de ventilación para acceder a la nave y activar allí el generador.

Objetivo 2. Subir a la azotea para hablar con Ryan.

Al pulsar el botón del ascensor se pondrá en marcha y el jugador ya podrás subir a la azotea y hablar por el ordenador público.

Objetivo 3. Ir a la entrada del parque.

Para terminar John tendrá de recorrer todo el nivel para ir a la entrada del parque, donde allí terminará el juego.

Objetivo 4 (OPCIONAL): El jugador tiene la opción de conseguir todos los créditos que hay por el nivel.

Objetivo 5 (OPCIONAL): El jugador tiene la opción de conseguir todos los coleccionables del juego, hay un máximo de diez.

3.6 Concept art

El juego tiene un diseño al estilo comic, tanto a la introducción como todos los elementos del juego, incluso se ha utilizado un estilo de letra acorde. La referencia a los diseños se basando principalmente del videojuego *Beyond a Steel Sky* para PC del año 2020. Adjuntamos capturas de algunas escenas del juego.



Ilustración 3. Imágenes del videojuego *Beyond a Steel Sky*

Adjuntamos también algunos bocetos realizados a mano para posteriormente diseñar y modelar los elementos finales del escenario.

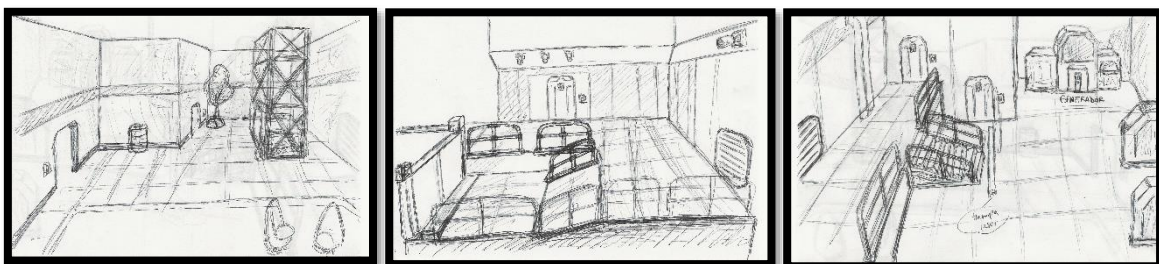


Ilustración 4. Bocetos de algunos edificios

4. Diseño técnico

4.1 Entorno desarrollo elegido



Después de evaluar varios los engines, me he decantado por el motor Unity por varias razones.

Este motor tiene mucha documentación disponible, la curva de aprendizaje es aceptable (Unreal tiene una dificultad alta), tiene su versión gratuita y totalmente completa sin limitaciones y por último me permite utilizar el lenguaje C# en la que ya tengo cierta experiencia y me ahorraré el aprendizaje del lenguaje.

4.2 Requisitos técnicos del entorno de desarrollo

Se ha utilizado **Unity Hub** con la versión Unity 2021.1.22
22 Sep. 2021. Los requisitos mínimos del sistema son:

- **SO:** Windows 7 SP1+, 8, 10, únicamente en las versiones de 64 bits; Mac OS 10.12 o posterior; Ubuntu 16.04 y CentOS 7.
- **GPU:** Tarjeta gráfica con capacidad DX10 (shader modelo 4.0).

Indicamos las características del equipo utilizado para la realización del proyecto:

Procesador: Intel(R) Core(TM) i7-4702MQ CPU @ 2.20GHz 2.20 GHz

SO: Windows 10 Home

RAM: 16 GB

Tarjeta gráfica: Nvidia GeForce GT 750M

4.3 Inventario y descripción de las herramientas utilizadas

Recursos necesarios utilizados para realizar el trabajo:



Sourcetree. Para subir los datos a GitHub.



Unity Hub. Es el motor 3D que utilizaremos para crear el juego.



Visual Studio Code. Para la programación de los scripts.



Adobe Illustrator 2021. Para dibujar los escenarios, personajes, etc.



Photoshop 2021. Para la edición de la imágenes y traspaso de formato.



Adobe Audition 2021. Para grabar o retocar los diálogos, música o efectos.





Adobe After Effects 2021. Para realizar los videos de la introducción del juego, el video del logo, acciones jugador, etc.



Adobe Media Encoder 2021. Para montar los videos creados con After Effects.

En el administrador de paquetes de Unity (**Package Manager**) hemos utilizado los siguiente:

- **2D Animation v.6.0.5.** Para crear los huesos del jugador y poder animar fácilmente.
- **2D PSD Importer v.5.0.4.** Para poder cargar el dibujo del jugador de Photoshop a Unity.
- **ProBuilder v.5.0.3.** Para crear fácilmente el modelado de los suelos, figuras del escenario del nivel y poner texturas a estos.
- **TextMeshPro v.3.0.6.** Para crear textos en el juego con mejor calidad y resolución de una forma optimizada.
- **Cinemachine v.2.7.8.** Para configurar fácilmente la cámara al jugador en el nivel.
- **Input System v.1.0.2.** Para configurar de forma sencilla los periféricos del teclado y gamepad.

4.4 Inventario y descripción de los assets y recursos al juego.

Personaje principal.

El personaje de John se ha diseñado con un boceto a mano de su cara y después se ha escaneado para dibujarlo con Illustrator .

En la parte del cuerpo hemos utilizado de referencia de una imagen de [Pinterest](#) [1] ya que nos aportaba diferentes perspectivas del personaje para poder dibujar de forma lateral, frontal y posterior. Con Illustrator hemos combinado todo cambiando un poco el modelo original y añadiéndole varias manos, bocas, ojos, arma para así

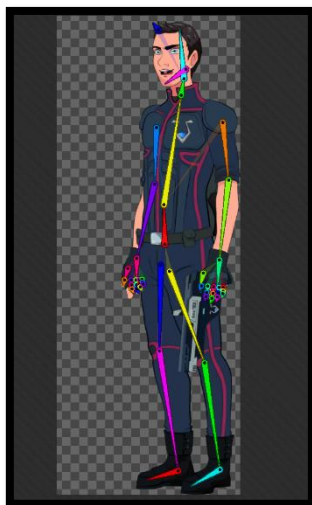


Ilustración 10. Personaje aplicándole huesos.

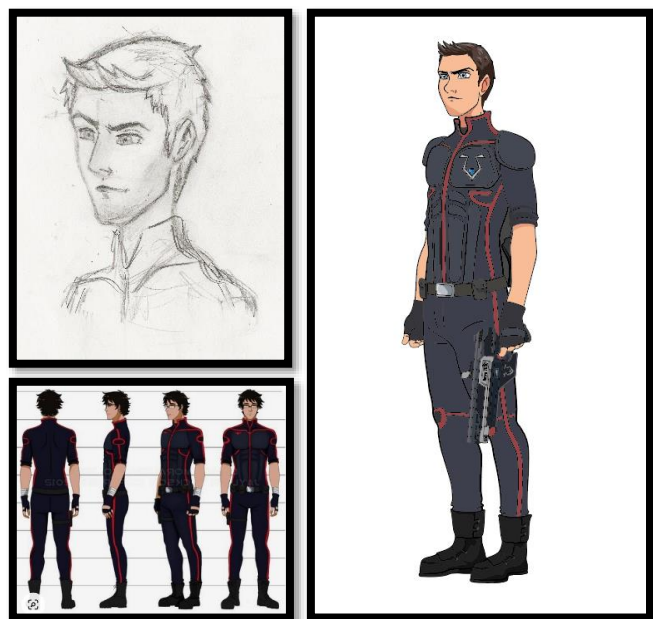


Ilustración 9. John Baker – Personaje principal

posteriormente poder animar el personaje. Cada una de esas partes se han puesto en varias capas distintas y se ha guardado en formato *.psb con Photoshop .

En Unity después de instalar el paquete [2D Animation](#) [8] y [2D PSD Importer](#) [9], nos hemos ido al Sprite editor de la imagen, le hemos añadido los huesos y se ha aplicado los pesos de influencia en la deformación de la malla del Sprite. Con esto ya hemos podido realizar cada una de las animaciones con el *Animator* moviendo o girando estos huesos del personaje en todo lo que dura cada clip de animación.

Personajes secundarios.

Se han diseñado con Illustrator **AI** varios personajes que aparecen en anuncios o son personajes 3D dentro del videojuego [Beyond a Steel Sky](#) [5]. Se han animado y distribuido en todo el escenario.

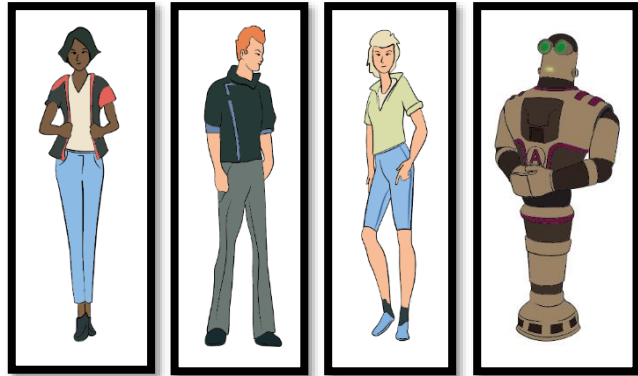


Ilustración 6. Diseño de personajes secundarios

Enemigos.



Ilustración 7. Diseño de enemigos

Se han creado con Illustrator **AI** tres tipos de enemigos a partir de capturas de varios personajes en 3D que aparecen en el videojuego [Beyond a Steel Sky](#) [5] cambiando algunos detalles. Se han distribuido en varias capas para después poder animar cada parte con sus movimientos y giros. Se crean varias luces de diferentes colores con degradado y transparencia para poder indicar en las animaciones los diferentes estados del robot.

Ítems.

Se han diseñado con Illustrator **AI** tres tipos de ítems de diseño propio. El primero lo hemos llamado "Lorguin", son coleccionables y hay un máximo de diez en todo el nivel. El segundo ítem son los créditos, que es dinero en el que se pueden comprar objetos útiles. El ultimo es un corazón para poder recuperar un porcentaje de vida perdida del jugador.

También con estos diseños se han creado tres texturas que hemos utilizado para los modelos en 3D.

En Unity **U** después de instalar el paquete [ProBuilder](#) [10], hemos modelado estos ítems y aplicado las texturas creadas.



Ilustración 13. Diseño ítems en 3D y de la UI

Interfaz objeto interactivo.



Ilustración 14. UI Objeto interactivo

Se ha creado con Illustrator **AI** la interfaz con un diseño propio y hay tres iconos para las acciones *ver*, *utilizar* y *hablar*. También se ha creado una caja de dialogo para añadir el texto indicando que se puede hacer con ese elemento interactivo. Hemos realizado una primera animación de solo el punto para indicar que es interactivo y si se pulsa un botón u otro aparecerá con otra animación con el texto del objeto que estamos interactuando y el icono de lo que podemos hacer.

Flechas indicativas de movimiento.

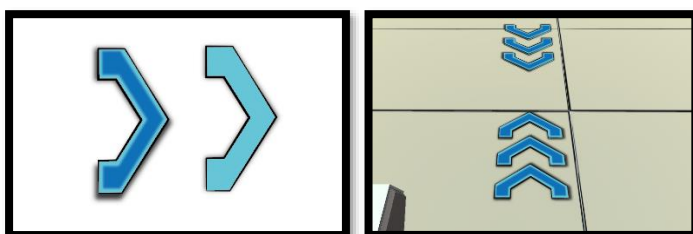


Ilustración 15. Flechas indicación movimiento jugador

Con Illustrator **AI** hemos hecho las flechas interactivas de diseño propio. En el *Animator* se animado con tres flechas que se van iluminado consecutivamente en bucle. Sirven para indicar al jugador que el jugador se puede desplazar para esa dirección.

Anuncios Publicitarios.



Ilustración 16. Diseño anuncios y soporte publicitario

Con Illustrator **AI** hemos realizado diez anuncios diferentes cogiendo imágenes que aparecen en el videojuego *Beyond a Steel Sky* [5] como modelo de referencia y cambiando el tipo de anuncio de este.

Con el *Animator* se ha creado una animación en la que se van alternando los anuncios de forma aleatoria gracias a su script. También se ha

modelado un soporte en 3D con *ProBuilder* [9] para el visionado de anuncios.

Efectos.

Con Illustrator **AI** a partir de dos imágenes originales de *humo* [2] y *circulo fuego* [3] como base se han redibujado de nuevo. Después hemos animado con *Animator* para poder usar estos efectos cuando se coge un ítem o en el título del juego en el menú principal.

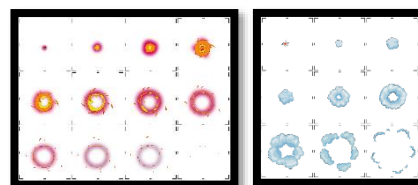


Ilustración 17. Efecto humo i circulo fuego

Elementos de Fondo de la escena.

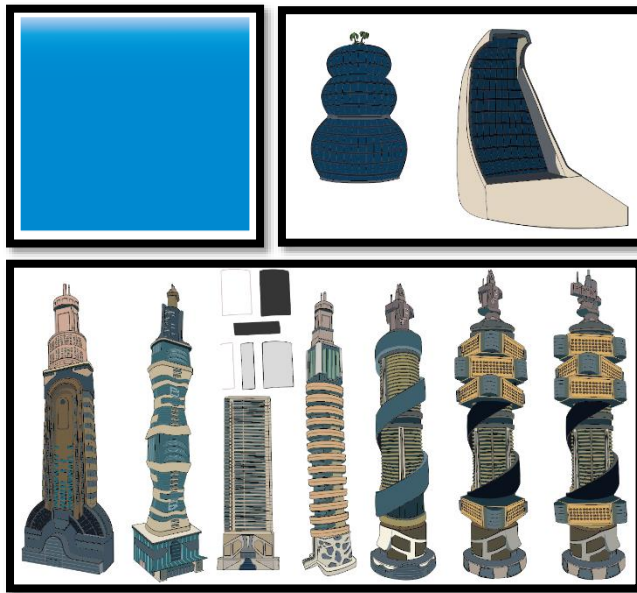


Ilustración 18. Diseño fondos escena del nivel

Con Illustrator **Ai** se ha cogido imágenes que aparecen en el videojuego [Beyond a Steel Sky](#) [5] como modelo de referencia para dibujar los rascacielos y algún edificio cambiando un poco el diseño. También se ha hecho un fondo azul para el cielo y unas nubes con difuminado para crear después con Animator una animación en las que se van moviendo en bucle.

Edificios del escenario.

Con Illustrator **Ai** se ha cogido imágenes que aparecen en el videojuego [Beyond a Steel Sky](#) [5] para crear edificios parecidos, pero con un diseño totalmente diferente. Después se han creado las paredes 3D para aplicar las texturas.



Ilustración 19. Diseño edificios del escenario

Elementos del escenario.

Se ha diseñado con Illustrator **Ai** cogiendo imágenes que aparecen en el videojuego [Beyond a Steel Sky](#) [5] para hacer las texturas y después en Unity **U** con [ProBuilder](#) [9] se ha modelado esos elementos como bancos, farolas, ordenador, etc. Algunos otros se han creado con diseño propio como las cajas, la estatua, el panel informativo, etc.

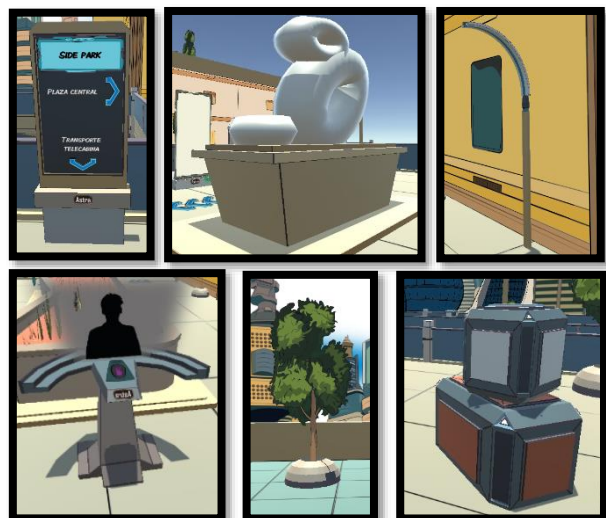


Ilustración 20. Diseño elementos del escenario

Pantallas opciones del juego.

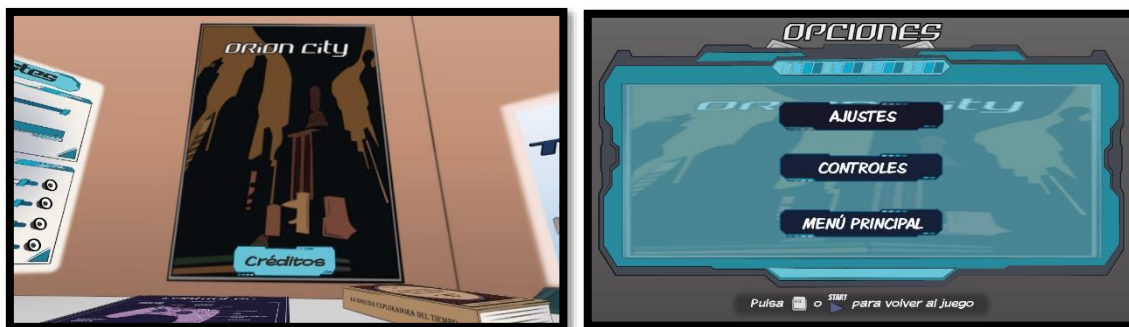


Ilustración 21. Diseño pantallas opciones

Con Illustrator **AI** cogiendo imágenes que aparecen en el videojuego [Beyond a Steel Sky](#) [5] se ha diseñado el despacho del menú principal, todo lo demás se ha realizado con un diseño propio.

Creación de videos.

Con After Effects **Ae** se han realizado los videos de introducción, los créditos, el logo y los videos con las interacciones del personaje activando objetos.

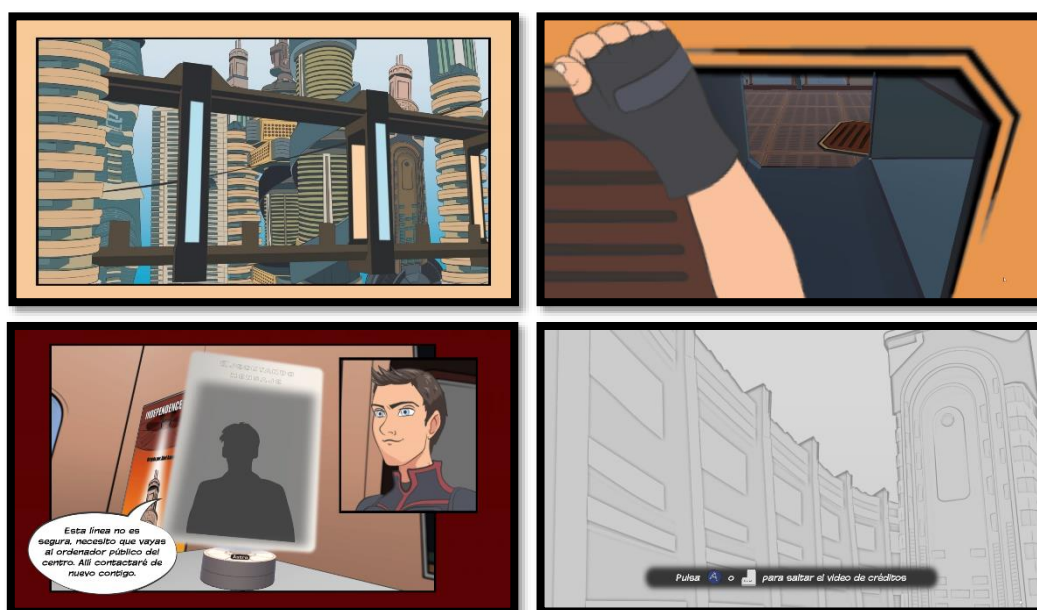


Ilustración 22. Pantallas de los videos del juego

El clip de la **introducción** esta formado por dos partes:

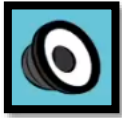
La primera el personaje principal nos introduce en la historia en la que nos describe la ciudad donde nos encontramos al estilo viñetas de comic. Además, aparece el titulo del juego y unos créditos iniciales.

La segunda parte mantiene el estilo viñeta, pero en este caso el personaje habla primero con su asistente de voz y después con otro personaje poniéndonos en situación de que es lo que tendremos de hacer solo empezar el nivel.


El clip de los **créditos** se han reaprovechado las mismas imágenes que salían en la introducción editándolas para después utilizarlas de fondo para los créditos que van saliendo.

Los clips de las **interacciones** del personaje, se han capturado los renders del propio juego y utilizado los brazos del jugador para realizar las animaciones. En total se han hecho cuatro videos.

Sonidos



Los sonidos utilizados tanto la música como los sonidos se han escogido de las librerías de audios de *Youtube studio* [4], las voces de diálogos, saltos y daño se ha utilizado nuestra propia voz.

Con Audition  se han gravado las voces y realizado modificaciones para crear efectos telefónicos o voces robóticas.

4.5 Esquema de arquitectura del juego/ componentes

Hay dos clases principales que se encargan de gestionar todo el juego y todas las demás al final se conectan con estas para pasarles información. Son las siguientes:

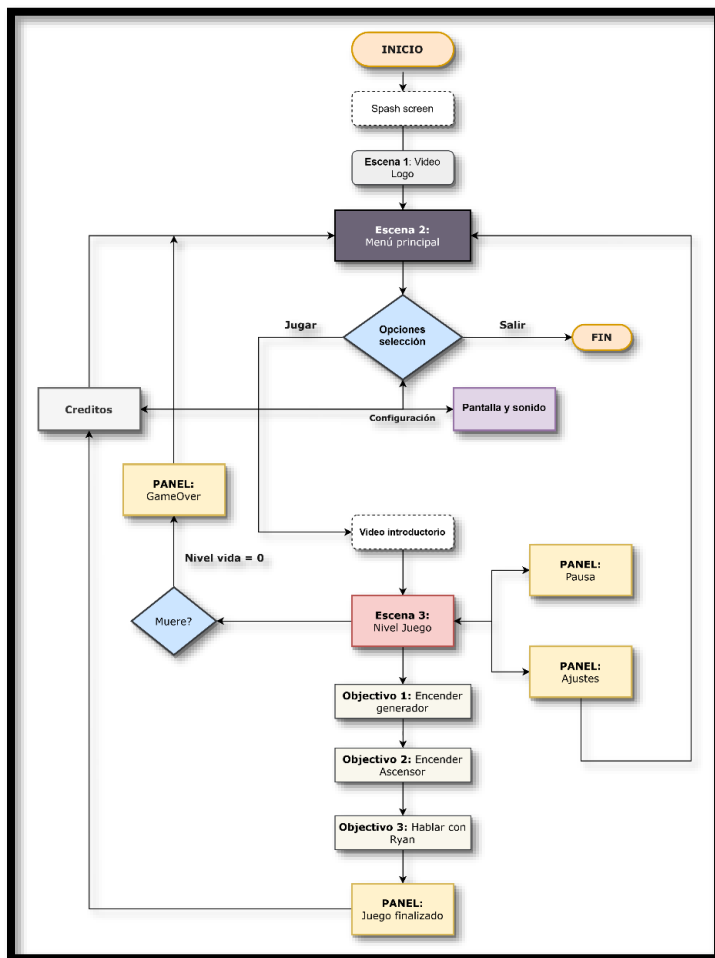


Ilustración 23. Diagrama de flujo del juego

AdministradorJuego.cs

Es la clase principal del juego, esta se encarga de almacenar por un lado las configuraciones de los ajustes con *PlayerPrefs* y además consulta constantemente si recibe información como que hemos conseguido un ítem o que nos ha hecho daño un enemigo, con lo que actualizara en la UI del nivel esos valores. También controla los objetivos del jugador si activa el interruptor del generador ya podrá subir al nivel y así sucesivamente. En ciertos momentos activara nuevos enemigos para que bloquear el paso al jugador.

ControladorJugador.cs

Es la clase que se encarga de controlar el jugador con *Input System* y aplicar sus movimientos, las fuerzas de salto, las interacciones con objetos y para ir a las opciones.

4.6 Funcionamiento de la IA de los enemigos



ControladorEnemigos.cs

Es la clase que gestiona a los enemigos y sus puntos de patrulla. Si el enemigo detecta al jugador en su campo de visión dejará el modo patrulla e irá a por él. En caso que el enemigo se haya alejado cierta distancia de su último punto de patrulla volverá su último punto para volver a patrullar. En caso que el enemigo se acerque bastante al jugador se le activara el modo atacando pegando al jugador para hacerle daño. Si consigue alcanzarle el enemigo, informará al *AdministradorJuego.cs* para que se actualice la barra de vida.

5. Diseño de niveles

El nivel ha sido diseñado por diferentes zonas, las cuales quedan bien definidas por los edificios que se encuentran en estas, además hay carteles informativos que indican cada espacio. Indicamos a continuación cada uno de ellos.

5.1 Apartamentos Tower5.



Ilustración 24. Diseño zona apartamentos

Aquí es donde el jugador empieza a jugar, concretamente desde el vestíbulo del edificio. En la plaza hay dos enemigos y varios ítems para poder coger. Además, el jugador recibirá varias instrucciones de ayuda sobre los enemigos o los ítems que vaya encontrando. En la parte izquierda inferior se encuentra la entrada al parque, que es el punto final del juego.

5.2. Plaza central.

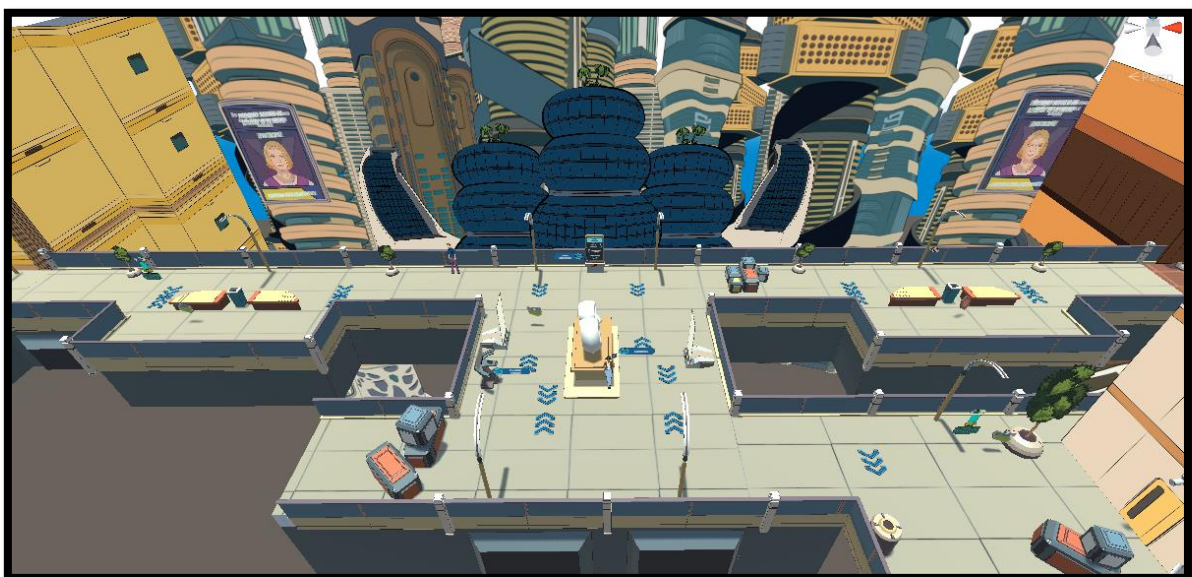


Ilustración 25. Diseño zona plaza central

La plaza central es una zona de paso para ir a los diferentes objetivos, aquí se encuentran varios enemigos y obstáculos para dificultar al jugador.

5.3 Torre mirador.



Ilustración 26. Diseño zona Torre mirador

En este edificio tenemos de averiguar cómo poder subir a la azotea por el ascensor, hay un enemigo y varios ítems para poder coger.

5.4 Central eléctrica.



Ilustración 27. Diseño zona Central eléctrica

En este edificio de la central eléctrica tendremos de entrar primero por la rejilla de ventilación, evitar unas trampas laser y dos enemigos con una IA básica para realizar el objetivo de encender el generador del fondo. También hay varios ítems que podremos coger.

6. Análisis de costes

Se ha calculado del coste total de este proyecto desglosándolo en la tabla siguiente y contando una duración para realizarlo de tres meses. El programa de Unity no se ha incluido como coste ya que es gratuito.

Costes	Comentario	Importe
Electricidad		215,71 €
Salario de un trabajador	2.000 euros/ mes (brutos)	6.000 €
Ordenador portátil		900 €
Monitor ordenador		150 €
Micrófono		110 €
Mando Xbox		40 €
Adobe Creative Cloud	Licencia para tres meses de proyecto	181,47 €
MS Project 2019 Professional	Pago único de una licencia	14,90 €
COSTE TOTAL		7.612.08 €

<i>Desglose coste electricidad</i>	
Potencia contractada	12,58€
Por energía consumida	180,13€
Impuesto electricidad	0,96€
Alquiler de contador	2,43 €
IVA normal 10%	19,61 €
Total importe factura	215,71 €

Se indica coste energético de los recursos consumidos en un periodo de tres meses según la comercializadora Endesa S.A.

Consumo de electricidad: 741,60 kWh

Para calcular las *emisiones* producidas hemos tenido en cuenta en el proyecto solo el consumo de eléctrico durante el cuarto trimestre. El factor de emisiones se ha consultado en el registro de huella de carbono, compensación y proyectos de absorción de dióxido de carbono para el año 2021 [5].

Electricidad: 0,25 kg CO₂/kWh (factor mix para al 2021)

Teniendo estos datos, resulta inmediato el cálculo de las emisiones para la efectucción del proyecto, indicamos el resultado en el siguiente cuadro.

Fuente	Dato actividad		Factor de emisión	Emisiones (kg CO ₂)
	Cifra	Unidad		
Electricidad	741,60	kWh	0,25	185,40

Por tanto, la huella de carbono para la realización del proyecto en tres meses ha sido de 185,40 kg CO₂.

7. Manual de usuario

7.1 Requerimientos mínimos técnicos del sistema.

- **SO:** Windows 7 SP1+, 8, 10.
- **RAM:** Mínimo 4 GB de RAM.
- **GPU:** Tarjeta gráfica con capacidad DX10 (shader modelo 4.0).
- **Disco duro:** Se requiere de 700 Mb de espacio en el disco.

7.2 Instrucciones del juego.

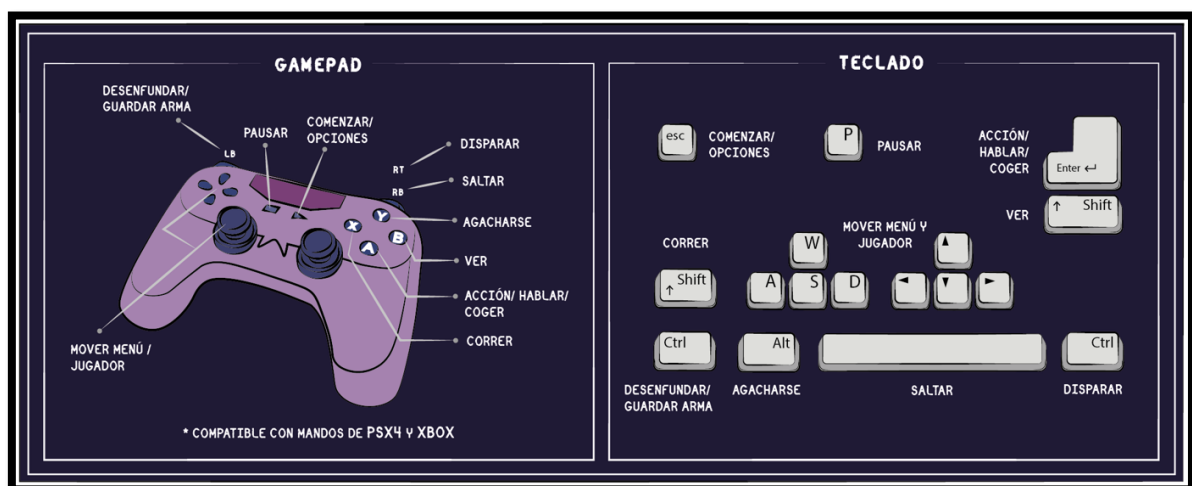


Ilustración 28. Controles del juego

Los controles para poder jugar son los indicados en la imagen. Las acciones de desenfundar/guardar arma y disparar se crearon diseño y la acción en el mapa de acciones del paquete de Input system pero no se llegaron a implementar por código.



Son las opciones para poder interactuar con los objetos interactivos. Esta la opción de Ver, Utilizar o Hablar.



Son los tres ítems que nos encontraremos en el escenario, son coleccionables, créditos y corazones para recuperar vida.



Son los tres tipos de enemigos que nos encontraremos en el nivel. Los dos primeros son especialmente los más peligrosos ya que si nos detectan nos atacarán.

8. Conclusiones

Como conclusión hemos aprendido de una forma extensa el funcionamiento el motor grafico de Unity para la creación y gestión de los assets y la programación de los scripts, la comunicación entre ellos y con los objetos asociados. También hemos aprendido a utilizar el paquete de ProBuilder para modelar, 2D Animator para crear huesos e Input system para la configuración de los periféricos. Con todo lo visto, ahora me veo con ganas de realizar nuevos proyectos y seguir aprendiendo incluso con nuevos géneros como de lucha o carreras.

En cuanto a los objetivos planteados inicialmente no se han podido cumplir, sobre todo porque se habían indicado un gran número de objetivos y funcionalidades que por el tiempo que teníamos era imposible de conseguir. Por eso, desde el inicio del desarrollo se redujeron a un mínimo y a partir de allí se fueron ampliando según el tiempo disponible y los contratiempos que fuimos encontrando al testear el videojuego.

La planificación se podido seguir correctamente en cada plazo ya que, como hemos dicho, se planteo al final un numero de objetivos básicos y a partir de allí se fueron ampliando nuevas localizaciones, enemigos o trampas según la disponibilidad que teníamos. En cada plazo se fueron corrigiendo errores y aun en esta ultima entrega no se han corregido algunos fallos visuales, mejoras en ayuda o de interacción, además de poner voces a los elementos interactivos.

Hay líneas de trabajo que no se han podido explorar o han quedado pendientes. De cara al futuro me gustaría realizar lo siguiente:

- Poder implementar la utilización del arma al jugador para disparar a los enemigos.
- Mejorar las animaciones de salto del jugador para darle más fluidez.
- Poder interactuar con más personajes para hablar con ellos.
- Tener la opción de coger objetos para utilizarlos posteriormente.
- Crear nuevas localizaciones como el parque o la zona de transporte.
- Realiza nuevos objetivos para poder ir destripando mas sobre esta historia de viajes temporales durante el juego.
- Ponerle nuevas trampas y enemigos al jugador para que sean más variados los niveles.

9. Glosario

Asset: es un elemento que puede ser usado en el proyecto o juego. Puede ser un modelo 3D, un archivo de audio, una imagen o cualquiera que sea compatibles con Unity.

C# (C Sharp): Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por la empresa Microsoft como parte de su plataforma .NET.

Canvas: Es el área donde todos los elementos UI deben estar.

Collider: Son áreas que se le añaden a un *GameObject* y se usan para que sea posible detectar colisiones entre distintos objetos del juego.

GameObject: son objetos fundamentales en Unity que representan personajes o elementos del escenario. Funcionan como contenedores para los componentes.

IA: Son las siglas para denominar la expresión de *inteligencia artificial*.

Indie: Es la abreviación del término en inglés *independent* que en español significa *independiente*.

Prefabs: Actúa como una plantilla a partir de la cual se pueden crear nuevas instancias del objeto en la escena.

Script: Son fragmentos de código que tienen como objetivo realizar o añadir funciones a un *GameObject*.

Sprite: Son imágenes de mapa de bits que al importarlas a Unity se convierten en objetos 2D.

Trigger: Es un disparador que en un objeto permite detectar si otro objeto ha entrado en su área de colisión.

UI: La expresión viene de *User Interface*, que significa *Interfaz del Usuario*. Es la vista que permite al usuario interactuar con el sistema.

10. Bibliografía

Assets:

- [1] JayJackDraws. 2012. *Danny's Final Turnaround* [imagen]. Pinterest.
Disponible en: <https://co.pinterest.com/pin/431993789251326458/>
- [2] Microone. - . *Puff, auge, estallido, explosión,...* [imagen]. 123RF.
Disponible en: https://es.123rf.com/photo_75443610_puff-auge-estallido-explosi%C3%B3n-explosi%C3%B3n-de-marcos-de-animaci%C3%B3n-de-vectores-de-efectos-boom-explota-a.html
- [3] Krol. - . *Sprites de energía de explosión o anillo,...* [imagen]. 123RF.
Disponible en: https://es.123rf.com/photo_58783130_sprites-de-energ%C3%ADa-de-explosi%C3%B3n-o-anillo-de-cuadros-de-animaci%C3%B3n-iconos-el-uso-en-el-desarrollo-del-.html
- [4] No requieren atribución. - . *The Awakening - Patrick Patrikios, Addicted – VYEN,...* [Sonidos]. YouTube Studio.
Disponible en: <https://studio.youtube.com/>

Interés:

- [5] **Revolution Software** [en línea] [fecha de consulta: 7 de Julio de 2021]. Disponible en: https://revolution.co.uk/games_catalog/beyond-a-steel-sky/
- [6] **Ministerio para la transición ecológica y el reto demográfico** [en línea] [fecha de consulta: 10 de diciembre de 2021]. Disponible en: <https://www.miteco.gob.es/es/cambio-climatico/temas/mitigacion-politicas-y-medidas/calculadoras.aspx>

Manuales y tutoriales:

- [7] **Unity** [en línea] [fecha de consulta: 12 de Agosto de 2021]. Disponible en: <https://docs.unity3d.com/Manual/index.html>
- [8] **Unity - 2D Animation** [en línea] [fecha de consulta: 30 de Agosto de 2021]. Disponible en: <https://docs.unity3d.com/Packages/com.unity.2d.animation@1.0/manual/index.html>
- [9] **Unity - 2D PSD Importer** [en línea] [fecha de consulta: 30 de Agosto de 2021]. Disponible en: <https://docs.unity3d.com/Packages/com.unity.2d.psdimporter@6.0/manual/index.html>
- [10] **Unity – ProBuilder** [en línea] [fecha de consulta: 10 de Julio de 2021]. Disponible en: <https://docs.unity3d.com/Packages/com.unity.probuilder@5.0/manual/index.html>
- [11] **Comp-3 Interactive**. (18 de noviembre de 2021). *How to play random animations [Unity Tutorial]* [vídeo en línea].
Disponible en: https://www.youtube.com/watch?v=MmrCUuwy-SE&ab_channel=Comp-3Interactive
- [12] **Pato Gimenez Punta**. (25 de noviembre de 2021). *Aprende a usar el nuevo Input System de Unity (2020)* [vídeo en línea]. Disponible en: https://www.youtube.com/watch?v=QezmoP3GblU&ab_channel=PatoGimenezPunta
- [13] **DGonzalez Tutoriales Unity en Español**. (10 de julio de 2021). *#20 - Fin de nivel. Cambios de escena completos con fadeout. [Unity, tutorial completo]* [vídeo en línea]. Disponible en: https://www.youtube.com/watch?v=suY381unmqE&list=PLBxM7h9DJBp5Bfg15JGf-kkXJw-1r6CPY&index=22&ab_channel=DGonzalezTutorialesUnityenEspa%C3%B1ol