

DarkFuturaVR

Autor: Jordi Pérez Sandonís
Máster Universitario en Diseño y Programación de Videojuegos
(Programación avanzada)

Tutores: Jordi Duch Gavalda
Helio Tejedor Navarro

Profesor: Joan Arnedo Moreno

05/06/2022

Créditos/Copyright



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

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>DarkFuturaVR</i>
Nombre del autor:	<i>Jordi Pérez Sandonís</i>
Nombre del colaborador/a docente :	<i>Jordi Duch Gavalda / Helio Tejedor Navarro</i>
Nombre del PRA:	<i>Joan Arnedo Moreno</i>
Fecha de entrega (mm/aaaa):	<i>06/2022</i>
Titulación o programa:	<i>Máster Universitario en Diseño y Programación de Videojuegos</i>
Área del Trabajo Final:	<i>TFM - Programación avanzada</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Realidad Virtual, Videojuegos, Trabajo Final de Máster</i>
Resumen del Trabajo	
<p>La finalidad de este proyecto es desarrollar un juego de Realidad Virtual para el visor Oculus Quest 2. Este desarrollo servirá tanto para aplicar conceptos y técnicas trabajadas en el máster como para aprender a desarrollar en una nueva plataforma que no se ha utilizado en el máster pero que está emergiendo de forma importante en el sector, la Realidad Virtual. El juego DarkFuturaVR sitúa al jugador en el interior de una estación espacial en la que deberá sobrevivir a un incidente solucionando situaciones adversas resolviendo puzzles y eliminando enemigos. La interacción se centrará en la manipulación de objetos 3D (agarrar, mover, lanzar, ...) y disparar a enemigos. Para el desarrollo se utilizará el motor Unity y el SDK de Oculus Quest que está disponible para este motor mediante un plugin. Repositorio: https://gitlab.com/kavra/darkfuturavr.git</p>	
Abstract	
<p>The purpose of this project is to develop a Virtual Reality game for the Oculus Quest 2 viewer. This development will serve both to apply concepts and techniques worked on in the master's degree and to learn how to develop on a new platform that has not been used in the master's degree but is emerging in an important way in the sector, Virtual Reality. The game DarkFuturaVR places the player inside a space station where he must survive an incident by solving adverse situations by solving puzzles and eliminating enemies. Interaction will focus on manipulating 3D objects (grabbing, moving, throwing,...) and shooting enemies. For the development, the Unity engine and the Oculus Quest SDK will be used, the last is available for the engine through a plugin. Repository: https://gitlab.com/kavra/darkfuturavr.git</p>	

Agradecimientos

A mi mujer Rosa y a mi hija Edna por su apoyo y comprensión durante el proceso de elaboración de este proyecto y durante el transcurso del máster. Sin estos ingredientes habría sido imposible finalizarlo. De verdad, muchas gracias.

Índice

1.Introducción.....	7
1.1.Introducción/Prefacio.....	7
1.2.Descripción/Definición.....	8
1.3.Metodología y proceso de trabajo.....	9
1.4.Planificación.....	11
2.Análisis de mercado.....	12
2.1.Público objetivo (<i>i.e. target audience</i>) y perfiles de usuario.....	12
2.2.Marco teórico/estado del arte.....	12
2.2.1.Usos de la Realidad Virtual.....	12
2.2.2.Movimiento en la Realidad Virtual.....	13
2.2.3.Interacción en la Realidad Virtual.....	14
2.2.4.Puntos de vista en la Realidad Virtual.....	14
3.Propuesta.....	15
3.1.Definición de objetivos/especificaciones del producto.....	15
3.2.Modelo de negocio.....	15
3.3.Estrategia de marketing.....	15
4.Diseño.....	16
4.1.Entorno de desarrollo.....	16
4.2.Herramientas de soporte.....	16
4.3.Assets.....	17
4.3.1.Gráficos (modelos 3D e imágenes).....	17
4.3.2.Sonido.....	17
4.4.Diseño de interacción y movimiento.....	17
4.4.1.Movimiento, posicionamiento y punto de vista.....	17
4.4.2.Interacción.....	18
4.5.Presentación de la información.....	18
4.6.Primer nivel.....	18
4.6.1.Estructura.....	18
4.6.2.Enemigos.....	20
4.7.Diagrama de navegación.....	20
5.Implementación.....	21
5.1.Requisitos de instalación.....	21
5.2.Instrucciones de instalación.....	21
5.3.Implementación de los diales (Reto 2).....	21

5.4.Implementación de los enemigos.....	21
5.5.Implementación del sistema de voces.....	22
5.6.El GameManager.....	22
6.Demostración.....	23
6.1.Tests.....	23
7. Conclusiones y líneas de futuro.....	24
7.1.Conclusiones.....	24
7.2.Líneas de futuro.....	25
8.Anexos.....	26
8.1.Solución a los retos.....	26
8.1.1.Reto1: Código de la puerta de la sala de hibernación.....	26
8.1.2.Reto2: Tres diales del puente de mando para desbloquear la puerta.....	26
8.2.Controles del juego.....	26
8.3.Enlaces producto y repositorio.....	27

1.Introducción

1.1. Introducción/Prefacio

En las últimas décadas el ocio electrónico, concretamente el sector de los videojuegos, ha experimentado un enorme crecimiento pasando de ser un sector minoritario en los años 70 del siglo XX a convertirse en el mayor sector de ocio en las primeras décadas del siglo XXI superando a la industria del cine en cuanto a público y volumen de negocio.

Durante estas décadas se ha visto una evolución técnica y tecnológica que ha dado lugar a la creación de numerosas plataformas de hardware en las cuales se han podido ejecutar videojuegos (videoconsolas domésticas de sobremesa y portátiles, máquinas arcade, ordenadores, etc.) además de nuevas formas de interacción de los jugadores con dichas plataformas.

El hardware se ha vuelto más potente, más pequeño y más eficiente ofreciendo cada vez mejores gráficos y sonidos y además se han creado numerosos sistemas portátiles que han dejado de depender de pantallas externas y conexión a enchufes para alimentarse durante su uso.

Las ruedas y botones que se utilizaron en los inicios de los videojuegos para interactuar con los sistemas evolucionaron a controladores más complejos, con distribuciones de botones más estandarizadas (cruquetas, agrupación de botones de acción en el frontal, latiguillos, ...), con palancas y botones sensibles a distintos niveles de presión (botones y palancas analógicos), se hicieron sensibles a las fuerzas físicas facilitando el reconocimiento de su movimiento por parte del sistema (Nintendo WiiMote). Actualmente se puede conocer la posición relativa de los controladores y su orientación e incluso se utilizan técnicas de reconocimiento de imagen que han permitido interactuar con los juegos sin necesidad de mandos como es el caso de las cámaras Kinect de Microsoft que reconocen los movimientos corporales o el visor Oculus Quest 2 que reconoce las manos y los gestos que pueden realizarse con los dedos.

Esta evolución ha traído a nuestros días los visores de realidad virtual, los cuales hacen que el jugador perciba los juegos de una forma totalmente inmersiva y además pueda interactuar con ellos de forma mucho más natural al hacer el seguimiento de sus extremidades. Actualmente esta tecnología está en auge, su cuota de mercado está creciendo y las inversiones están creciendo enormemente debido no solo a sus aplicaciones en el sector del ocio sino que también a sus aplicaciones en muchos otros sectores como la industria, la educación, la psicología, etc.

1.2. Descripción/Definición

La finalidad de este proyecto final de máster es crear un producto de entretenimiento de realidad virtual que se ejecute en un visor representativo del mercado actual sin la necesidad de utilizar un ordenador externo y explorar las posibilidades que ofrecen este tipo de dispositivos.

El objeto a desarrollar durante el proyecto es un juego que sitúe al jugador en un espacio tridimensional y le permita moverse e interactuar con el entorno utilizando sus manos/mandos de una forma natural. El jugador debe poder agarrar, empujar y lanzar objetos, realizar disparos con armas y presionar botones presentes en el escenario.

La historia del juego se desarrollará en el interior de una nave espacial en la que el protagonista deberá sobrevivir a un conjunto de situaciones adversas originadas por un incidente que afectará a la seguridad de la nave espacial. La acción transcurrirá en espacios interiores pequeños y el acceso entre unos espacios y otros podrá estar bloqueado hasta que el jugador resuelva un puzzle o enigma o siga las instrucciones especificados por el juego. Habrá zonas en las que el jugador deberá utilizar armas para eliminar unos enemigos que se habrán adentrado en la nave espacial y amenazarán su supervivencia.

1.3. Metodología y proceso de trabajo

El visor de realidad virtual para el que se realizará el juego objeto de este proyecto es el Oculus Quest 2. La elección de dicho dispositivo se debe a que es un dispositivo standalone, tiene una alta cuota de mercado y es relativamente asequible en comparación al resto de dispositivos de este tipo.

Para elaborar un producto de este tipo existen dos planteamientos distintos:

1. Crear un motor de videojuegos completo que cubra las exigencias del juego
2. Elaborar el producto utilizando motor existente

Las restricciones de tiempo (4 meses) y recursos (un solo desarrollador) hacen que elaborar un motor de videojuegos con capacidades para Realidad Virtual desde la base no sea una opción viable. Es por ello que se ha decidido utilizar un motor existente en el mercado que se ajuste a las necesidades del proyecto.

Los motores que se han tenido en cuenta para el desarrollo del videojuego tienen versiones gratuitas para este tipo de proyectos, incluyen las herramientas necesarias para crear el tipo de videojuego deseado y además permiten utilizar el SDK de Oculus Quest mediante un plugin. Estos motores son:

- Unity (<https://unity3d.com>)
- Unreal Engine (<https://www.unrealengine.com>)

Ambos motores son perfectamente válidos para realizar este proyecto ya que técnicamente no tienen ninguna limitación que impida utilizarlos. Hay algunos criterios que decantan la balanza hacia uno en particular, Unity. Este motor es el más utilizado en el desarrollo de videojuegos indie, la curva de aprendizaje es menos pronunciada que en Unreal Engine, hay muchos más recursos (documentación, foros, tutoriales, ...) disponibles en la red, su comunidad es más extensa y el lenguaje que se utiliza para programarlo es C#, de más alto nivel y más sencillo que el utilizado por Unreal Engine (C++). Además, de los dos motores, Unity es el único que se ha utilizado en las asignaturas del máster, por lo cual no se tendrá que aprender a utilizarlo para hacer el proyecto a diferencia de si se utilizara Unreal Engine.

Debido a la limitación temporal del proyecto, se ha decidido limitar el tiempo dedicado a la elaboración de recursos gráficos (modelos 3D y texturas) y recursos sonoros (música y efectos de sonido) para dedicar el mayor tiempo posible a la programación de mecánicas, la elaboración del escenario y el diseño de la jugabilidad. Para ello se utilizarán recursos elaborados por terceros adquiridos en sus respectivas tiendas o en tiendas de bundles.

El diseño de una experiencia o juego de realidad virtual debería tener como elemento central al jugador.

Deben tenerse en cuenta los siguientes factores

1. El jugador debe percibir correctamente y entender la información que se muestra
2. El jugador debe entender y saber ejecutar las interacciones que requiere el juego
3. Los movimientos y posturas necesarias para jugar deben ser ergonómicas
4. Las acciones deben ser sencillas de ejecutar
5. Las imágenes y sonidos deben ser agradables al usuario a menos que el diseño requiera lo contrario

-
6. Se deben incluir los elementos necesarios para que el jugador no se encuentre perdido o sin saber qué hacer

Atendiendo a los puntos anteriores se considera que durante la fase de diseño deberían aplicarse conceptos o mecanismos propios del diseño centrado en el usuario (DCU) ya que estos ayudarán a detectar problemas que puedan encontrar los usuarios en fases prematuras del proyecto y el diseño final será más sólido.

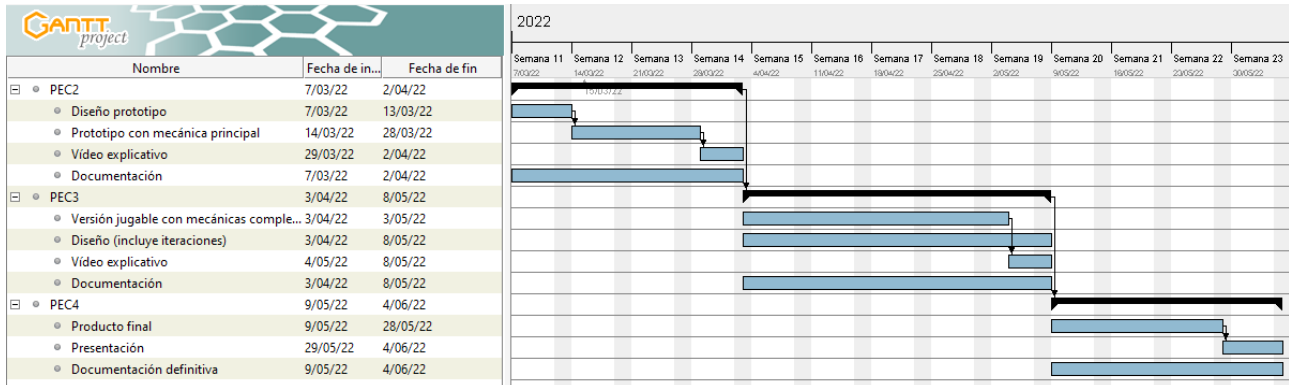
Durante la fase de implementación, para cada funcionalidad implementada, se verificará a través de pruebas si los resultados obtenidos cumplen con los requisitos especificados. En caso contrario se tendrán en cuenta los problemas detectados y se procederá a una nueva iteración. Cuando el resultado sea el deseado se procederá a implementar la siguiente funcionalidad.

1.4. Planificación

El tiempo que se dedicará al proyecto dependerá del día de la semana:

- 2h en días laborables
- 4h en días festivos

resultando en 18h semanales.



2. Análisis de mercado

2.1. Público objetivo (*i.e. target audience*) y perfiles de usuario

La definición de un público objetivo y los perfiles de usuario para el juego que se presenta en este trabajo ha presentado ciertas dificultades ya que la información estadística sobre usuarios de realidad virtual ha resultado ser poco abundante y muy sesgada.

En primer lugar no se han encontrado estadísticas exactas sobre el número de jugadores de realidad virtual ni estudios sobre perfiles de jugadores de esta plataforma, géneros, etc. Solo se ha encontrado información sobre el número de visores vendidos de algunos modelos o la cuota de un determinado visor en la plataforma Steam (<https://store.steampowered.com/hwsurvey>) y esta información no ha resultado útil para este caso.

Debido a esto se ha tomado la decisión de aproximar las características del público objetivo de este producto a las características de los jugadores de productos similares en PC y videoconsolas de última generación.

El público al que está dirigido este producto es una persona de entre 12 y 50 años, que utiliza la tecnología a diario, muy aficionada a los videojuegos en PC y/o videoconsolas, con preferencia por los juegos del género de acción / aventuras y con un nivel adquisitivo medio / alto.

2.2. Marco teórico/estado del arte

2.2.1. Usos de la Realidad Virtual

A continuación se describen algunos de los usos que se dan actualmente a la realidad virtual:

1. Simulación de conducción / pilotaje. Consiste en situar al jugador en los mandos de un vehículo y hacerle sentir la experiencia de pilotaje / conducción. Además del visor pueden utilizarse distintos tipos de mandos y controles como palancas, botones, pedales o volantes y la experiencia puede ampliarse con el uso de asientos o plataformas que añaden movimiento y vibración aportando mayor inmersión.
2. Tele-presencia. Consiste en situar al jugador en un lugar virtual en el que tiene lugar un evento o espectáculo convirtiéndolo en un espectador o participante. Ejemplos de ello son un cine virtual, una conferencia o presentación virtual, un concierto virtual, etc.
3. Soporte en psicología. Consiste en utilizar la realidad virtual como herramienta de soporte en terapias psicológicas. Ejemplos de ello son experiencias que exponen de forma controlada a los pacientes frente a sus miedos, por ejemplo mostrando arañas para combatir la aracnofobia,

situando al paciente ante un grupo de avatares para combatir el miedo a hablar en público o mejorar las habilidades, etc.

4. Soporte en Educación. Consiste en utilizar simulaciones, visualizaciones de escenas y mundos interactivos para mejorar el aprendizaje y aumentar la atención y el interés de los alumnos.
5. Mundos virtuales, Metaverso. La posibilidad de generar y experimentar mundos virtuales se explota tanto para mundos imaginarios en los que rigen leyes físicas que nada tienen que ver con la realidad como mundos que pretenden emular la realidad física que nos rodea. La emulación del mundo real está sirviendo para probar diseños de fábricas en sus dimensiones reales de forma virtual pudiendo ser recorridas por personas e incluso realizar actividades en su interior como si de la fábrica real se tratase, lo que permitiría detectar problemas de diseño antes de iniciar la construcción real. Actualmente, este uso de la realidad virtual está creciendo mucho en volumen económico gracias a empresas desarrolladoras como Meta (antigua Facebook) o Nvidia.
6. Videojuegos. Consiste en situar al jugador en la acción de un videojuego de forma inmersiva dándole la posibilidad de observar el entorno con el movimiento de su cabeza e interactuar con el entorno utilizando el movimiento de sus extremidades.
7. Experiencias contemplativas. Consiste en situar al jugador en una posición estática desde la que puede contemplar distintos tipos de imágenes y paisajes acompañados de sonidos y música pensados principalmente para contribuir a la relajación y el bienestar.
8. Fitness. Consiste en ofrecer al jugador la posibilidad de realizar actividades que demandan movimiento y esfuerzo físico en un entorno virtual. Un ejemplo son los juegos de golpear con las manos a los objetos que se dirigen al jugador, juegos de combates de boxeo, etc.
9. Investigación. Actualmente la realidad virtual se utiliza en estudios sobre conducta humana, sensibilización (poniendo a personas en la situación de alguien que sufre bullying o acoso), etc.

2.2.2. Movimiento en la Realidad Virtual

Las distintas experiencias y juegos desarrollados para realidad virtual han presentado varias maneras de mover al jugador y la cámara dentro de los mundos virtuales.

1. Giro gradual. Es una manera de girar la cámara en el plano horizontal que emula el giro lateral de la cabeza con la particularidad que el giro se realiza por tramos de varios grados (30, 45, 90, ...). Este giro se realiza accionando una palanca o botón del controlador o mando del jugador.
2. Giro natural. A diferencia del caso anterior, en este caso el giro de la cámara replica el giro de la cabeza del jugador.
3. Teletransporte. Consiste en desplazar al jugador de forma instantánea a un lugar que previamente haya marcado con un cursor.

4. Hotspots. Se dispone de un punto o cursor situado en el centro de la visión del jugador, si este punto se mantiene sobre un punto predefinido del entorno virtual, este lanza un evento que puede servir para mostrar información u otro tipo de comportamiento programado por los desarrolladores.
5. Sobre raíles. El jugador se desplaza por un raíl invisible con la única opción de cambiar de recorrido en puntos concretos predefinidos. Un ejemplo claro son las experiencias y juegos de montañas rusas.
6. Natural locomotion. Es un sistema que reconoce ciertos movimientos de las extremidades del jugador gracias a los acelerómetros de los controladores y los traduce en órdenes de desplazamiento, salto, etc.
7. Movimiento con mando/controlador. El movimiento del jugador se especifica con los controles direccionales (sticks, palancas o botones) del mando/controlador.
8. Movimiento real. El desplazamiento y movimientos reales del jugador se trasladan al mundo virtual. Este tipo de desplazamiento requiere de un área de juego amplia y libre de obstáculos para disfrutar de la experiencia de forma segura.

2.2.3. Interacción en la Realidad Virtual

La realidad virtual generalmente presenta mundos interactivos en los que el jugador puede realizar un conjunto de acciones contra los objetos virtuales. Estas acciones se realizan con las manos, representadas en el mundo virtual mediante avatares. El posicionamiento de las manos virtuales se realiza mediante el seguimiento de las manos reales o los mandos / controladores. Los gestos de las manos o movimientos de los dedos pueden realizarse accionando botones de los mandos / controladores o pueden ser capturados a través de las cámaras de los visores si estos disponen de esta capacidad. Las interacciones más usuales en la realidad virtual son:

1. Agarrar / soltar / lanzar / golpear / mover objetos
2. Accionar botones / mover palancas
3. Apuntar y disparar

2.2.4. Puntos de vista en la Realidad Virtual

1. Primera persona. La cámara se sitúa en la posición del jugador que ejerce de protagonista en la acción. El jugador se mueve e interacciona con el entorno virtual de forma directa. Este tipo de cámara es el típicamente utilizado en los juegos de acción como "Half-Life: Alix".
2. Tercera persona. La cámara se sitúa alejada del protagonista de la acción enfocándole en todo momento. El jugador indica lo que debe realizar el protagonista, interacciona con el entorno o simplemente observa. Este tipo de cámara se utiliza en el juego "Moss".

3.Propuesta

3.1. Definición de objetivos/especificaciones del producto

El producto a realizar en este proyecto es una experiencia inmersiva de realidad virtual que pretende entretener al jugador, hacerlo pensar y llevarlo a un lugar distinto a la realidad de su día a día. Las características principales del juego son:

1. Situar al jugador en una nave espacial, viviendo una aventura en primera persona.
2. Incorporar retos similares a los que se encuentran en los juegos de escape (deducir un código, manipular artilugios mediante botones/palancas/ruedas,...)
3. Incorporar elementos de los juegos de disparos en primera persona

3.2. Modelo de negocio

El modelo de negocio se basará en una compra de producto completo en un único pago en tienda online que permita disfrutar del juego completo para siempre o alternativamente, la venta del juego por capítulos o expansiones que requieran pagos adicionales.

3.3. Estrategia de marketing

Estudiando un poco los precios de los juegos de desarrolladores independientes o pequeños estudios se estima que un precio adecuado estaría por los 10€, siendo razonable 9.99€ o incluso bajando a los 6.99€ en caso de rebajas. Un precio superior, por ejemplo 20-30€ podría perjudicarlo si se compara este juego desarrollado de forma independiente con otros títulos en este rango de precios realizados por estudios experimentados. Tampoco sería una buena estrategia ponerle un precio de salida más bajo ya que podría hacer pensar que el juego no tiene suficiente calidad y de ahí viniera el precio tan bajo.

De caras al lanzamiento del juego sería interesante preparar una web con un diseño atractivo donde se muestren vídeos y capturas de los puntos fuertes del juego y contratar anuncios en alguna plataforma de advertising que se mostraran directamente al público objetivo del producto.

4. Diseño

4.1. Entorno de desarrollo

El núcleo del entorno de desarrollo empleado en este proyecto está comprendido por las tres piezas de software que se detallan a continuación:

- Unity 2020.3.30f1 (LTS). Motor de videojuegos multiplataforma que incluye un conjunto de herramientas completo para desarrollar videojuegos (editores de niveles, UI, animación, iluminación, IA, etc...). Permite compilar ejecutables para distintas plataformas y es ampliable mediante plugins.
- Microsoft Visual Studio Community 2019. IDE de programación gratuito de Microsoft que contiene todo lo necesario para programar en C# y depurar los proyectos. La programación del proyecto se ha realizado íntegramente con esta herramienta.
- Oculus Integration SDK para Unity. Plugin de integración de las herramientas de Oculus (Meta) para creación de experiencias de realidad virtual para visores de Oculus (Meta) utilizando el motor de creación de videojuegos Unity.

4.2. Herramientas de soporte

Además de las herramientas de creación mencionadas en el apartado anterior, ha sido necesario integrar al flujo de trabajo otras herramientas de soporte para la creación y manipulación de contenido digital:

- Inkscape 0.91: herramienta de dibujo vectorial. Se ha utilizado para crear el título y el splash screen del inicio del juego además de otros gráficos sencillos.
- Gimp 2.8: herramienta de dibujo y manipulación de imagen. Se ha utilizado para generar algunos efectos en el título del juego.
- Audacity 2.4.2: herramienta de grabación y edición de sonido multiplataforma de código abierto. Se ha utilizado para modificar assets de sonido.
- GitLab: servidor Git para gestionar los cambios durante el desarrollo de proyectos de mediana y gran envergadura. Ha permitido llevar un control de las modificaciones y revertir los cambios en situaciones anómalas.
- GitGUI 0.21: herramienta de control de versiones multiplataforma de código abierto. Se ha utilizado para trabajar con el repositorio de GitLab.
- LibreOffice 5.2.4.2: suite ofimática libre que se ha utilizado para generar la memoria de este proyecto.

4.3. Assets

En la fase inicial del proyecto se decidió utilizar principalmente assets de terceros para evitar la carga de trabajo que implica su creación y centrar los esfuerzos en el diseño y desarrollo del software objeto de este proyecto.

4.3.1. Gráficos (modelos 3D e imágenes)

Los assets gráficos de terceros utilizados en este proyecto son los siguientes:

- POLYGON - Sci-Fi Space Pack. Paquete de assets low poly para crear escenas de temática de ciencia-ficción espacial. Incluye naves espaciales, personajes, decoración, armas y módulos de construcción. <https://syntystore.com/products/polygon-sci-fi-space-pack>
- POLYGON - Sci-Fi City Pack. Paquete de assets low poly para crear entornos y ciudades de temática de ciencia-ficción . Incluye personajes, decoración, armas, vehículos y módulos de construcción. <https://syntystore.com/collections/frontpage/products/polygon-sci-fi-city>

Algunos de los assets contenidos en los paquetes mencionados tuvieron que ser alterados par cumplir con su objetivo en el juego y han dado lugar a nuevos assets con distinta estética o funcionalidad según el caso. Los nuevos assets se encuentran en la ruta /Assets/JPS del proyecto distribuidos en subcarpetas.

4.3.2. Sonido

Las voces del juego se han generado con la herramienta de conversión de texto a voz online gratuita Free TTS. <https://freetts.com/>

4.4. Diseño de interacción y movimiento

Al tratarse de una experiencia de realidad virtual, se ha considerado prioritario que el usuario actúe en el mundo virtual con la naturalidad del mundo físico concediéndose ciertas licencias en pro de la seguridad del jugador y de facilitar el uso del producto.

4.4.1. Movimiento, posicionamiento y punto de vista

El punto de vista/observación que presenta el juego es la primera persona, el jugador encarna al protagonista de la acción.

El jugador se puede mover en el espacio virtual de forma natural (caminando) o utilizando la palanca del controlador izquierdo. El movimiento vertical de la palanca lo desplazará hacia adelante y hacia atrás. El movimiento horizontal de la palanca lo desplazará lateralmente.

El jugador puede girar de forma natural (moviendo la cabeza o girando su cuerpo) o utilizando la palanca del controlador derecho. El movimiento horizontal de la palanca provocará un giro de 30° en la dirección que se indique.

La posición vertical de los ojos del avatar emulará la posición adoptada por el jugador en el mundo físico (de pie, agachado, tumbado, ...). La dirección a la que apunta la cámara es la indicada por la orientación de la cabeza del jugador.

4.4.2. Interacción

El juego presenta las manos virtuales del jugador en la posición y orientación de los mandos que sostiene en sus manos. Le será posible utilizarlas para interactuar con los objetos del juego (agarrar, soltar, empujar, ...) con cierta naturalidad. Para facilitar la interacción se ha optado por utilizar un solo botón (botón lateral del mando, dedo corazón) para las acciones de agarrar/soltar e interactuar con botones virtuales. Presionando dicho botón la mano virtual se cerrará y a la vez extenderá el dedo índice. Si un objeto "agarrable" está en el rango de acción de la mano será agarrado y en el momento que se deje de presionar el botón se soltará el objeto. Los objetos agarrados pueden ser lanzados y cambiados de mano. Cuando el dedo está extendido se podrán presionar botones virtuales de forma natural. Para utilizar la pistola deberá utilizarse el botón lateral para agarrarla y el botón del gatillo (dedo índice) para disparar.

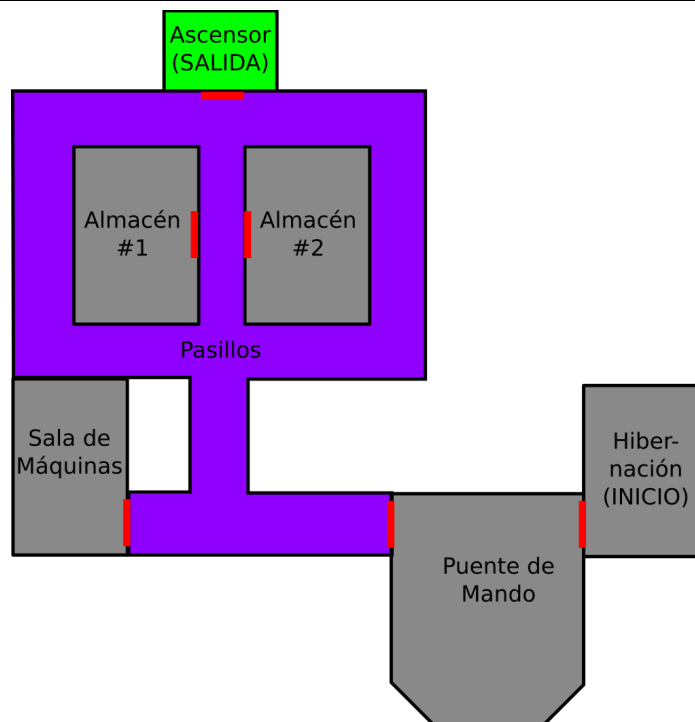
4.5. Presentación de la información

Durante la experiencia de juego, el jugador recibirá información sobre los objetos presentes en la escena y la evolución del juego. Se ha decidido minimizar la cantidad de texto que aparece en el visor sustituyéndolo por mensajes sonoros/vocales cuando se trate de mensajes largos y utilizar los mensajes flotantes de texto sólo en contextos muy concretos en los que se indique el nombre o información de un objeto presente en la escena.

4.6. Primer nivel

4.6.1. Estructura

El primer nivel del juego se desarrolla en una nave/estación espacial y está dividido en 5 habitaciones y varios pasillos:



Espacio	Descripción
Sala de hibernación	Inicio de la partida. Habitación con las cápsulas de hibernación de los capitanes. Reto: <ol style="list-style-type: none"> 1. Descubrir la combinación que debe introducirse en el panel de apertura de la puerta. Hay una pista en una PDA.
Puente de mando	Centro neurálgico de la nave. Retos: <ol style="list-style-type: none"> 1. Manipular el mecanismo de la pared para quitar el primer bloqueo de la puerta
Sala de máquinas	Contiene la maquinaria necesaria para alimentar varios sistemas de la nave. Retos: <ol style="list-style-type: none"> 1. Encontrar varios dispositivos energéticos y colocarlos en el puerto de carga para alimentar los sistemas auxiliares.
Pasillos	Comunican los espacios del nivel. Retos: <ol style="list-style-type: none"> 1. Eliminar los enemigos que se cuelan en la nave. 2. Trasladar dispositivos energéticos.

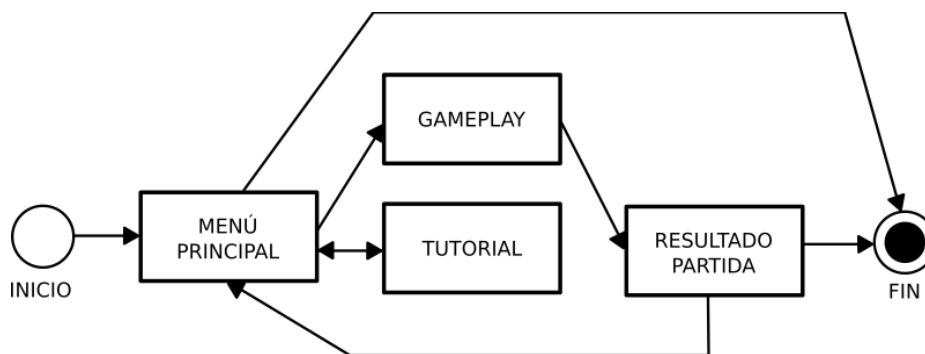
Almacén #1	Vacío y cerrado.
Almacén #2	Vacío y cerrado.
Ascensor	<p>Punto final del nivel. Reto:</p> <ol style="list-style-type: none"> 1. Eliminar la última oleada de enemigos. Éstos tendrán un comportamiento más agresivo que los drones vistos anteriormente en los pasillos.

4.6.2. Enemigos

En el primer nivel del juego solo se presenta un tipo de enemigo. Se trata de drones extraterrestres que intentarán eliminar al jugador y tomar el control de la nave. Estos enemigos aparecen de forma espontanea y están dotados de una inteligencia artificial simple basada en patrullar zonas hasta que detecten al jugador y entonces perseguirlo y atacarlo con disparos. Antes de finalizar el juego aparece una última oleada de enemigos con capacidades superiores:

1. detectan al jugador a mayor distancia
2. cuando detectan al jugador no dejan de perseguirlo
3. mayor velocidad de giro

4.7. Diagrama de navegación



5. Implementación

5.1. Requisitos de instalación

DarkFuturaVR requiere del visor de realidad virtual Meta Quest 2 (anteriormente Oculus Quest 2). Al no ser una aplicación publicada en las stores oficiales requiere la instalación mediante aplicaciones externas.

5.2. Instrucciones de instalación

Para la instalación correcta del ejecutable (apk) en el visor se recomienda utilizar alguno de los programas de escritorio disponibles en internet con soporte oficial:

- SideQuest: La primera tienda alternativa y aplicación de escritorio que permite instalar aplicaciones en el visor. <https://sidequestvr.com/>
- Oculus Developer Hub: Aplicación de escritorio oficial de Meta para desarrolladores que permite instalar aplicaciones en el visor, monitorizar el sistema y realizar otras operaciones mediante comandos de adb. https://developer.oculus.com/documentation/unity/ts-odh/?locale=es_ES

Véanse las páginas oficiales de las aplicaciones para encontrar información sobre como operar con ellas.

5.3. Implementación de los diales (Reto 2)

El problema que presenta este tipo de controles es que utilizando el sistema de agarre de objetos del plugin de Unity, éstos se hacen hijos de la mano, dando lugar a que su posición y rotación siga a las de la mano. Este es un comportamiento no deseado ya que el botón debe quedarse en su lugar de origen y solo debe girar respecto a su eje central. Para implementar los diales (o botones rotatorios) del reto del puente de mando se ha creado un cubo invisible aparte del modelo del dial y se utiliza a modo de ayuda. Este cubo invisible está encima del dial y es el objeto que realmente agarra el jugador. Al manipular el objeto, un script interpreta la rotación en el eje deseado y la transmite al objeto del dial. Para evitar que el dial gire cuando el jugador se lleva el objeto de soporte a más distancia que la deseada, el script que gobierna el dial hace que el cubo invisible se desligue de la mano y vuelva a su lugar de origen.

5.4. Implementación de los enemigos

El movimiento de los drones enemigos se ha implementado mediante el componente NavmeshAgent de Unity, el cual hace que los objetos puedan navegar por la escena gracias a una malla de navegación que se ha generado en tiempo de diseño y cubre los suelos de todo el nivel. Para evitar que los agentes atraviesen las puertas, éstas contienen un componente NavmeshObstacle. Para la inteligencia artificial de los enemigos se ha optado por implementar una máquina de estados que aplica un comportamiento distinto en función del estado en que se encuentra el enemigo. Los estados son los siguientes:

1. Patrullar. Se dirige a diferentes puntos de la escena almacenados en una lista. El orden es aleatorio. Si el jugador está dentro de su alcance pasa al estado de alerta.
2. Alerta. Realiza una vuelta de reconocimiento sobre su eje vertical y comprueba, mediante un raycast, si el jugador está delante suyo. En cuyo caso pasa al estado de ataque.
3. Ataque. Se encara directamente al jugador y dispara obedeciendo a la cadencia de disparo que se le ha especificado.
4. Berserker. Detecta al jugador a mayor distancia, aumenta su velocidad y persigue al jugador cuando lo detecta. En este estado solo se encuentran los enemigos de la última oleada.

La generación de enemigos la realiza un script llamado DroneManager, éste se encarga de generar los enemigos cuando sea necesario, en el lugar adecuado (contiene una lista con estos puntos), e indica a los enemigos los waypoints que deben utilizar para patrullar (contiene una lista con estos puntos). Este objeto es totalmente configurable mediante parámetros. Para facilitar la lectura en la escena se ha provisto de ayudas visuales a los waypoints y a los puntos de respawn mediante gizmos de Unity, concretamente esferas de distintos colores.

5.5. Implementación del sistema de voces

El sistema de voces y avisos del juego está implementado en un script llamado VoicesManager que contiene dos listas, una de AudioSource que especifica el origen del sonido y otra de AudioClip que contiene referencias a todos los clips de sonido de las voces. Éstas voces y avisos simulan ser locuciones de la inteligencia artificial de la nave y pensamientos del protagonista del juego y pretenden guiar al jugador y meterlo en la historia del juego. El sistema se vale de colliders invisibles que actúan a modo de disparadores y contienen una referencia al mensaje que debe invocarse.

5.6. El GameManager

El script que gobierna la lógica principal del juego es el GameManager. Se ha implementado mediante el patrón singleton y se ha dotado de métodos estáticos para que sea accesible desde cualquier lugar del juego. Algunas de las responsabilidades de este componente son el seguimiento del correcto cumplimiento de las reglas del juego (cumplimiento de hitos) y consecución de acontecimientos, la afectación de la salud del jugador y su muerte.

6. Demostración

6.1. Tests

El proceso de testeo y refinamiento durante el desarrollo de un producto para realidad virtual es más pesado y complejo que el de un juego de ordenador. Tener que compilar, transferir el ejecutable al dispositivo, ponerse el dispositivo y probar el juego sin poder manipular directamente/cómodamente el debugger o visualizar la consola de debug de forma sencilla hicieron que se tomara la decisión de realizar dos tipos de tests distintos:

1. Pruebas desde el editor de Unity (con o sin la ayuda del debugger). Todas las funcionalidades que pudieran probarse sin necesidad del visor se realizaron de este modo.
2. Pruebas con el visor. Todas las funcionalidades que requerían el seguimiento del visor o los mandos, la manipulación compleja de objetos o la comprobación del correcto funcionamiento de la iluminación y el sonido que fueran sensibles a funcionar distinto en el visor se realizaron directamente sobre el visor.

7. Conclusiones y líneas de futuro

7.1. Conclusiones

- Conclusiones del trabajo:
 - El diseño de experiencias para VR requiere de un estudio específico muy orientado a facilitar su uso pensando en la posición en la que se jugará, el espacio disponible y las capacidades físicas de los jugadores.
 - El testeo de experiencias para VR es más dificultoso que en otras plataformas como el ordenador debido a que los visores no disponen de las mismas herramientas de depuración o éstas no son tan cómodas de utilizar como en PC.
 - En las fases de diseño e implementación de experiencias para VR deben tenerse muy en cuenta las limitaciones del hardware y deben conocerse las mejores técnicas de optimización. Esto puede marcar la diferencia entre una experiencia difícil de correr en el hardware y una experiencia fluida que además sea un deleite para los sentidos.
- Una reflexión crítica sobre el logro de los objetivos planteados inicialmente:
 - Los objetivos generales que se plantearon al inicio del proyecto han quedado cubiertos, aunque el número de interacciones que puede realizar el jugador son inferiores a las que se se plantearon inicialmente. Ocurre lo mismo con el número de armas, el número de objetos, los tipos de enemigos y la cantidad y calidad de los retos que se plantean al jugador.
 - Los principales motivos del incumplimiento parcial de los objetivos han sido los siguientes:
 - Se subestimó el tiempo necesario para diseñar y testear las funcionalidades
 - Se subestimó el tiempo necesario para diseñar la experiencia de usuario
 - La dedicación al proyecto se vio reducida debido a circunstancias ajenas al proyecto
- Un análisis crítico del seguimiento de la planificación y metodología a lo largo del proyecto:
 - Durante la ejecución del proyecto han aparecido algunas desviaciones de varios días que se han ido manteniendo de forma estable hasta la fecha de entrega.
 - La metodología prevista ha sido adecuada y los problemas durante el desarrollo en ningún caso han sido derivados del método de trabajo.

-
- Como se ha comentado anteriormente se ha reducido la cantidad de interacciones, objetos y retos que se esperaba desarrollar inicialmente debido al aumento de tiempo dedicado al diseño de la experiencia de usuario y la imposibilidad de dedicar más tiempo al proyecto.

7.2. Líneas de futuro

El producto presentado en este proyecto, aunque es la demostración de un juego incompleto, y carece de ciertas funcionalidades, tiene una base robusta sobre la cual se puede terminar de construir el juego completo. Es por ello que se plantea continuar el juego para realizar un lanzamiento comercial en el futuro. Las principales líneas de actuación para mejorar y ampliar el juego son las siguientes:

1. Aumentar el entretenimiento añadiendo mecánicas, situaciones y retos que afecten al ritmo del juego y creen en el jugador la necesidad de seguir jugando.
2. Configuración del sonido para utilizar un sistema de audio posicional y aplicar modelos sonoros basados en la geometría de la escena que permitan aplicar oclusión, reverberación y atenuación del sonido para aumentar la inmersión del jugador en la experiencia de realidad virtual.
3. Mejorar el rendimiento y calidad de la iluminación aplicando técnicas precálculo de la iluminación global generando lightmaps en los objetos estáticos de la escena y utilizar lightprobes para aplicar iluminación a los elementos móviles de la escena.
4. Simplificar los colliders de la escena para reducir el cálculo de las físicas y aumentar el rendimiento.
5. Ampliar el número de tipos de enemigos y mejorar la inteligencia artificial haciendo que los enemigos tengan patrones de comportamiento más complejos como la capacidad de colaborar entre ellos para realizar emboscadas.
6. Añadir nuevas formas de interacción que permitan utilizar/manipular nuevos elementos como palancas, volantes, ruedas o manivelas.
7. Ampliar el número de objetos y armas que puede utilizar el jugador.
8. Añadir un sistema de inventario que permita llevar varios objetos sin necesidad de llevarlos en las manos y dé la posibilidad de poder utilizarlos en cualquier momento.

8. Anexos

8.1. Solución a los retos

8.1.1. Reto1: Código de la puerta de la sala de hibernación

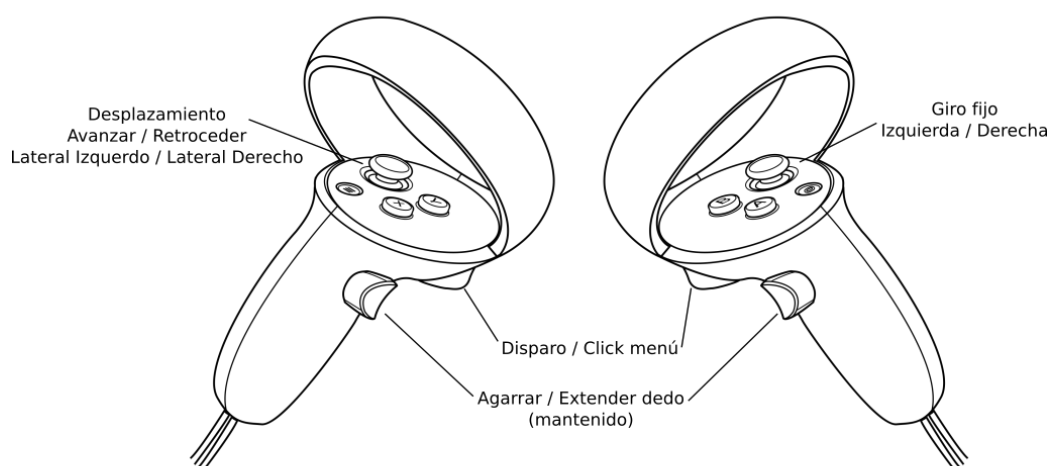
El jugador debe marcar en la botonera la combinación que abre la puerta. La PDA indica los números 2-1-2, éstos indican que en la primera fila de la botonera deben marcarse dos botones, en la segunda fila 1 botón y en la tercera fila 2 botones. Los botones que deben quedar marcados en azul son:

1. primera fila: primero y segundo
2. segunda fila: segundo
3. tercera fila: segundo y tercero

8.1.2. Reto2: Tres diales del puente de mando para desbloquear la puerta

El jugador debe conectar dos cubos del mismo color utilizando la barra del mismo color. La manipulación de las barras se realiza girando los tres diales o ruedas del artillugio. El segundo y tercer dial son de fácil deducción. El del medio sube y baja la barra seleccionada, el de la derecha cambia su inclinación. El primer dial permite seleccionar una barra u otra y sus tres posiciones están comprendidas en rangos de 120 grados de vuelta. Para comprobar su funcionamiento el jugador deberá probar las otras ruedas cambiando la posición de la primera hasta que se dé cuenta que girándola bastante cambia la barra seleccionada.

8.2. Controles del juego



8.3. Enlaces producto y repositorio

Ejecutable: https://drive.google.com/drive/folders/1k1D4xSOGW5ibYSDhf5bduM_pLB_BNLa?usp=sharing

Memoria: <https://drive.google.com/drive/folders/1iuA9FiHVIs8Aww4YONUhpBNZMIAm5e0z?usp=sharing>

Videos: <https://drive.google.com/drive/folders/1AUcroITUQQzwBPrrspy-c8d7z5FOXqIXZ?usp=sharing>

Repositorio: <https://gitlab.com/kavra/darkfuturaavr.git>

Release en repositorio: <https://gitlab.com/kavra/darkfuturaavr/-/releases/PEC4>

Ejecutable en repositorio:

https://gitlab.com/kavra/darkfuturaavr/-/raw/master/DarkFuturaVR/Ejecutable_APK/DarkFuturaVR.apk