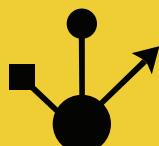




Escuela
Politécnica
Superior

Escape Rooms temáticos usando Realidad Virtual para prevenir Deterioro Cognitivo



Grado en Ingeniería Multimedia

Trabajo Fin de Grado

Autor:

Lucila Vázquez Soriano

Tutor/es:

Jose García Rodríguez

David Mulero Pérez

Diciembre 2024



Universitat d'Alacant
Universidad de Alicante

Escape Rooms temáticos usando Realidad Virtual para prevenir Deterioro Cognitivo

Autor

Lucila Vázquez Soriano

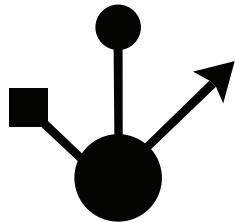
Tutor/es

Jose García Rodríguez

Departamento de Tecnología Informática y Computación

David Mulero Pérez

Departamento de Tecnología Informática y Computación



Grado en Ingeniería Multimedia



Escuela
Politécnica
Superior



Universitat d'Alacant
Universidad de Alicante

Abstract

In this project, a themed escape room has been developed using virtual reality technology to prevent and slow down cognitive deterioration. It is founded on two basic projects developed in Unreal Engine 4 for virtual reality: the first one¹, which developed an advanced hand-tracking system, and the second one², which, based on the previous one, developed a realistic object-grasping system. This technology has been the basis used in this work for interaction with elements and the virtual environment. Throughout the work, research was carried out on the global situation of cognitive impairment and how it is increasingly affecting an ageing population. Cognitive deterioration, characterised by a progressive decline in mental abilities such as memory, attention and problem-solving, affects the quality of life of those who suffer from it. In this context, preventive strategies have become an area of great interest, with particular focus on the use of innovative technologies. This project explores the creation of a virtual reality escape room as a preventive tool, combining the benefits of gamification with challenges designed to stimulate the cognitive abilities of the user.

Finally, a tracking system was developed to monitor the performance of players, which stores metrics in game time to later store them in files that can be read and reviewed by a professional to assess the user's situation.

¹<https://github.com/3dperceptionlab/unrealhandgrasp>

²<https://github.com/3dperceptionlab/unrealtrackinggrasp>

Resumen

En este proyecto se ha desarrollado un *escape room* temático utilizando tecnología de realidad virtual para prevenir y frenar el deterioro cognitivo. Se parte de dos proyectos base desarrollados en Unreal Engine 4 para realidad virtual: el primero ¹, que desarrolló un seguimiento de manos avanzado, y el segundo ², que a partir del anterior, desarrolló un sistema de agarre realista de objetos. Esta tecnología ha sido la base empleada en el trabajo para la interacción con los elementos y el entorno virtual.

A lo largo del trabajo se investigó sobre la situación global del deterioro cognitivo, y cómo, cada vez más, afecta a una población cada vez más envejecida. El deterioro cognitivo, caracterizado por una disminución progresiva de las capacidades mentales como la memoria, la atención y la resolución de problemas, afecta significativamente la calidad de vida de quienes lo padecen. En este contexto, las estrategias preventivas se han convertido en un área de gran interés, destacando especialmente el uso de tecnologías innovadoras. Este proyecto explora la creación de un escape room en realidad virtual como herramienta preventiva, combinando los beneficios de la gamificación con desafíos diseñados para estimular las capacidades cognitivas del usuario.

Por último, se desarrolló un sistema de seguimiento para monitorizar el desempeño de los jugadores, el cual almacena métricas en tiempo de juego para más adelante almacenarlas en ficheros que podrán ser leídos y revisados por un profesional para evaluar la situación del usuario.

¹<https://github.com/3dperceptionlab/unrealhandgrasp>

²<https://github.com/3dperceptionlab/unrealtrackinggrasp>

Agradecimientos

Me gustaría expresar mi gratitud al Departamento de Tecnología Informática y Computacional, por idear esta propuesta tan interesante que se ajustaba tanto a mis intereses. Por brindarme conocimiento y las herramientas necesarias para llevarlo a cabo. Quiero agradecer a José García, a Pablo Ruiz y especialmente a David Mulero por sus consejos y paciencia a lo largo de todo el desarrollo.

Por otro lado, quiero agradecer a mis amigos, y a todas sus distintas formas de motivarme a completar este proyecto. Agradecer también a mi familia por su apoyo incondicional. Por último, agradecer al primero que estuvo ahí, desde el día cero creyendo en mí. Sin ninguno de ellos habría llegado tan lejos como hoy.

A ti, mamá.

It's the job that's never started as takes longest to finish.

J.R.R. Tolkien.

Índice general

Abstract	v
Resumen	vii
Agradecimientos	ix
Lista de Acrónimos y Abreviaturas	xxi
1 Introducción	1
1.1 Motivación	1
1.2 Propuesta y Objetivos	1
1.3 Planificación	2
2 Estado del Arte	5
2.1 Estudio de la problemática	5
2.2 Deterioro Cognitivo	6
2.2.1 Estrategias Preventivas y Fomento de la Reserva Cognitiva	7
2.3 Gamificación	8
2.4 Realidad Virtual	9
2.4.1 Evolución de la Realidad Virtual	10
2.4.1.1 Orígenes históricos de la Realidad Virtual	11
2.4.1.2 Avances tecnológicos en el siglo XX	11
2.4.1.3 Auge de la Realidad Virtual en el entretenimiento	12
2.4.1.4 Renacimiento de la Realidad Virtual	12
2.4.1.5 Aplicaciones contemporáneas	13
2.5 Escape Room	13
2.5.1 Historia del Escape Room	14
2.6 Antecedentes	18
3 Metodología	21
3.1 Introducción	21
3.2 Hardware	21
3.2.1 Oculus Quest	22
3.2.2 Meta Quest 2	23
3.2.3 Meta Quest 3	24
3.2.4 Equipo para el desarrollo	25
3.3 Software	27
3.3.1 Unreal Engine 4	27
3.3.2 Sistema de agarre y Framework de hand tracking	28
3.3.3 Blender	28

3.3.4 Overleaf	29
4 Desarrollo	31
4.1 Introducción	31
4.1.1 Game Instance	32
4.2 Sistema de Puzzles	32
4.2.1 PuzzleManager	32
4.2.2 BP_PuzzleObject	33
4.2.3 BP_Puzzle	33
4.2.4 Flujo de ejecución de puzzles	34
4.3 Sistema de interacción	35
4.3.1 BP_InteractiveObject	35
4.3.2 BP_PlacementPoint	36
4.3.3 BP_ObjectPlacement	36
4.3.4 Funcionamiento en tiempo de ejecución/Flujo de ejecución	37
4.4 Candado numérico	38
4.5 Material Highlight	39
4.6 Pistas	39
4.7 Sonido	40
4.8 Reloj	41
4.9 Sistema de Seguimiento	42
4.9.1 Integración de Blueprints y C++	42
4.9.2 Implementación en C++	42
4.9.3 Estructuras de almacenamiento de datos	43
4.9.4 Gestión de datos y creación del archivo de registro	45
4.9.5 Gestión de niveles y almacenamiento de datos	46
4.9.6 CreateJSON	46
4.9.7 Estructura JSON	47
4.9.7.1 Construcción de la estructura JSON	47
4.9.8 Métodos de toma de medidas	48
5 Diseño del Escape Room	51
5.1 Introducción	51
5.2 Diseño del nivel	52
5.3 Diseño de puzzles	53
5.3.1 Puzzles	56
5.4 Entorno	73
5.5 Elementos	79
5.6 Reminiscencia	79
6 Conclusiones	81
6.1 Conclusiones	81
6.2 Trabajo futuro	82
Bibliografía	83

Lista de Figuras

2.1	Esperanza de vida global por países en 2023	5
2.2	Invento del estereoscopio y su uso posterior	11
2.3	Avances en el siglo XX en la tecnología de la realidad virtual	12
2.4	Realidad virtual en el mundo del entretenimiento	13
2.5	Captura del juego Zork	14
2.6	Capturas de los juegos <i>Mystery house</i> y <i>The Pawn</i>	15
2.7	Captura del juego Planet Mephius	15
2.8	Elementos del videojuego <i>Behind Closed Doors</i>	16
2.9	<i>MOTAS (2001)</i>	17
2.10	Takao Kato posando dentro de “The Real Escape Game”	17
2.11	algo aquí de los juegos	18
2.12	Antecedente de un <i>escape room</i> para Realidad Virtual (RV) <i>Escape Simulator</i> [30]	19
2.13	Antecedentes de juegos con puzzle y seguimiento de manos	19
2.14	Experiencia <i>escape room</i> en residencia en Ciempozuelos	20
3.1	Gafas y controladores Oculus Quest	23
3.2	Gafas y controladores Meta Quest 2	24
3.3	Gafas y controladores Meta Quest 3	25
4.1	Ejemplo de candado numérico dentro del <i>escape room</i>	38
4.2	Resultado del efecto Highlight para los objetos interactivos	39
4.3	Ejemplo de una pista sobre el puzzle de poner la mesa 5.3	40
4.4	Reloj en tiempo real dentro del juego	42
4.5	Ejemplo de resultado del fichero JavaScript Object Notation (JSON) generado.	50
5.1	Tipos de organización de los puzzles	52
5.2	Aspecto final del puzzle receta del nivel cocina (Figura 5.17)	57
5.3	Aspecto final del puzzle torre de magdalenas del nivel cocina (Figura 5.17) . .	58
5.4	Aspecto final del puzzle poner la mesa dentro del nivel cocina (Figura 5.17) . .	59
5.5	Aspecto final del puzzle patos de goma	60
5.6	Aspecto final del puzzle de palabras encadenadas del nivel baño (Figura 5.15) en su versión fácil	61
5.7	Aspecto final del puzzle secuencia de botes del nivel baño (Figura 5.15)	62
5.8	Aspecto final del puzzle de planificación para ducharse del nivel baño (Figura 5.15) sin resolver	63
5.9	Aspecto final del puzzle figuras 3 Dimensions (3D) del nivel salón (Figura 5.16)	65
5.10	Aspecto final del puzzle conexión de refranes del nivel salón (Figura 5.16 . . .	66
5.11	Aspecto final del puzzle colección de libros del nivel salón (Figura 5.16)	68
5.12	Aspecto final del puzzle del joyero del nivel alcoba (Figura 5.18)	69

5.13 Aspecto final del puzzle del vestirse del nivel alcoba (Figura 5.18)	71
5.14 Aspecto de la puerta para salir del nivel alcoba	72
5.15 Diseño final de la estancia báño desde distintos ángulos	74
5.16 Diseño final de la estancia salón desde distintos ángulos	75
5.17 Diseño final de la estancia cocina desde distintos ángulos	76
5.18 Diseño final de la estancia alcoba desde distintos ángulos	77
5.19 Diseño final de la estancia recibidor desde distintos ángulos	78
5.20 Colección de objetos propios	79
5.21 Elementos diseñados para la terapia de reminiscencia	80

Lista de Tablas

3.1	Especificaciones para usar Unreal Engine	26
3.2	Especificaciones para usar Meta Quest Link	26
3.3	Especificaciones del ordenador personal usado	27
4.1	Variables en BP_PuzzleObject	33
4.2	Variables en BP_Puzzle	34
4.3	Variables en BP_InteractiveObject.	36
4.4	Variables en BP_PlacementPoint.	36
4.5	Parámetros de CreateLogFile	43
4.6	Parámetros de WriteToFile	43
4.7	Parámetros de ReadFromFile	43
4.8	Campos de Object Interactions Structure	44
4.9	Campos de Headset Structure	44
4.10	Campos de Level Structure	45
5.1	Detalles del puzzle receta	56
5.2	Detalles del puzzle torre de magdalenas	57
5.3	Detalles del puzzle de poner la mesa	58
5.4	Detalles del puzzle patos de goma del nivel baño	59
5.5	Detalles del puzzle palabras encadenadas	60
5.6	Detalles del puzzle secuencia de botes	62
5.7	Detalles del puzzle planificación de ducharse	63
5.8	Detalles del puzzle vistas de figuras	64
5.9	Detalles del puzzle de refranes	65
5.10	Detalles del puzzle colección de libros	67
5.11	Detalles del puzzle del joyero y sus joyas	68
5.12	Detalles del puzzle de vestirse	70
5.13	Detalles del puzzle final de cada nivel	72

Lista de Acrónimos y Abreviaturas

2D	2 Dimensions.
3D	3 Dimensions.
CC	Capacidad Cognitiva.
CPU	Central Processing Unit.
DLC	Deterioro Cognitivo Leve.
DOF	Degrees Of Freedom.
DTIC	Departamento de Tecnología Informática y Computación.
EPS	Escuela Politécnica Superior.
FPS	Fotogramas por segundo.
Gbps	Gigabits por segundo.
GPU	Graphics Processing Unit.
JSON	JavaScript Object Notation.
LCD	Liquid Crystal Display.
MCI	Mild Cognitive Impairment.
OLED	Organic Light Emitting Diode.
PC	Personal Computer.
RAM	Random Access Memory.
RV	Realidad Virtual.
TFG	Trabajo Final de Grado.
UE	Unreal Engine.
UE4	Unreal Engine 4.
UE5	Unreal Engine 5.
VR	Virtual Reality.

1 Introducción

En los últimos años no han dejado de aumentar las ventas de dispositivos de Realidad Virtual (RV), ni de crecer la inversión por parte de multitud de empresas en esta tecnología [6]. Además, la realidad virtual está llegando a nuevos campos. Aunque se asocia con el entretenimiento digital, lo cierto es que se va abriendo hueco entre otros sectores [3, 35], y surge la necesidad de seguir investigando y desarrollando aplicaciones para esta tecnología.

1.1 Motivación

La primera vez que escuché hablar de la realidad virtual fue durante mi adolescencia, a través de una serie de animación llamada *Sword Art Online*. En aquel momento, la idea de adentrarme en un mundo de fantasía me parecía algo irreal, sacado directamente de un cuento, una posibilidad tan distante que apenas podía creerla posible. Años más tarde, tras leer *Ready Player One*, esa distopía comenzó a parecerme no tan lejana.

Llegado el momento de decidir qué estudiar tras las pruebas de acceso a la universidad, me encontré indecisa hasta el último día, balanceando pros y contras sin poder elegir. Fue entonces cuando, leí sobre la carrera de “ingeniería multimedia”, hasta hoy en día no entiendo por qué no me interesé por ella antes, sabía que quería estudiar una carrera de la rama de ingenierías y arquitectura. Decidí investigar su plan de estudios, y entonces lo vi: en el cuarto curso, una asignatura llamada “realidad virtual”. Fue gracias a esas dos palabras que escogí esta carrera.

Al estudiar esa asignatura, comencé a ver la realidad virtual no solo como una herramienta de entretenimiento, sino como un portal hacia infinitas posibilidades, permitiéndome entrar en mundos que antes solo había podido imaginar leyendo. Recientemente había jugado a un videojuego de temática *Scape Room*, fue entonces cuando pensé en la posibilidad de desarrollar un *Scape Room* para RV. Cuando vi este proyecto entre las propuestas de Trabajo Final de Grado (TFG), no solo parecía la opción ideal, sino que también añadía un propósito valioso: algo que no solo supondría un desafío y me permitiría aprender sobre uno de los motores más utilizados en la industria, sino que mejoraba la idea pudiendo contribuir a la sociedad. Supe inmediatamente que tenía que intentar que me lo asignasen.

1.2 Propuesta y Objetivos

El objetivo principal de este TFG es desarrollar una serie de *Escape Rooms* empleando tecnología de RV. Diseñados con la finalidad de prevenir el deterioro cognitivo en personas mayores y facilitar la rehabilitación de pacientes con dicho deterioro. Para lograr esto, utilizaremos la tecnología desarrollada en el proyecto UnrealTrackingGrasp [25], desarrollado

por el 3DPLab de la Universidad de Alicante. Este proyecto es una continuación de los desarrollos previos UnrealHandGrasp [21] y UnrealGrasp [23], los cuales han permitido la eliminación del uso de controladores mediante el seguimiento avanzado de nuestras manos. Permitiéndonos, de esta forma, únicamente necesitar estas últimas a la hora de interactuar dentro del entorno virtual, proporcionando una experiencia de inmersión total para el usuario.

Para apoyar el objetivo principal, hemos dividido el proyecto en distintos puntos a realizar a lo largo del proyecto:

- Investigación sobre el Deterioro Cognitivo: Realizar un estudio exhaustivo sobre el deterioro cognitivo, su evolución y, sobre todo, sus contramedidas.
- Estudio del Mercado y Posibilidades: Analizar el mercado actual para encontrar inspiración y obtener ideas que puedan influir en el diseño de los *escape rooms*.
- Capacitación en la Tecnología Utilizada: aprender para ser capaz de utilizar tanto el motor gráfico a utilizar como poder sacarle provecho al proyecto del que se parte.
- Diseño de Niveles y Puzzles: Planificar la estructura y los desafíos de cada escape room para asegurar que sean adecuados y estimulantes para nuestro público objetivo.
- Creación de un Sistema de Puzzles: Desarrollar un sistema flexible y adaptable que permita personalizar los puzzles y agilizar su creación eliminando así repetición innecesaria de código.
- Programación de los Puzzles: Implementar dentro del motor los puzzles diseñados en el entorno de VR.
- Creación de Escenas: Diseñar ambientes virtuales que sean coherentes y estimulantes pero a su vez familiares para el público objetivo, contribuyendo a la experiencia general del escape room.
- Modelado y Texturizado de Elementos de los Puzzles: Modelar y texturizar elementos visuales y/o interactivos que sean parte integral de los puzzles con el objetivo de evocar reminiscencias en los usuarios.
- Desarrollo de un Sistema de Seguimiento: Implementar un sistema que permita monitorizar el avance del usuario por los *escape rooms*.

Estos objetivos están diseñados para asegurar que cada aspecto del proyecto contribuya de manera efectiva a la meta de proporcionar una herramienta terapéutica valiosa para nuestros usuarios, al mismo tiempo que se explora el potencial de las tecnologías de VR en el campo de la salud cognitiva.

1.3 Planificación

Para el desarrollo de este proyecto contaré con el respaldo del equipo de 3DPLab del Departamento de Tecnología Informática y Computación (DTIC) de la Universidad de Alicante. El desarrollo de este proyecto se puede dividir en 4 fases.

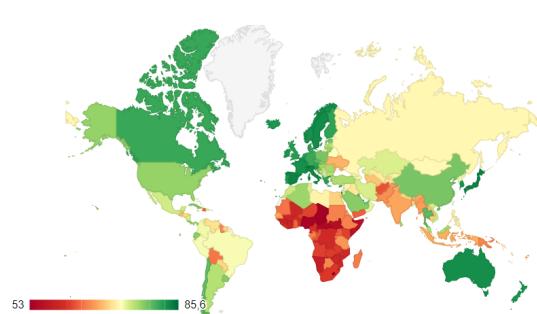
- Primera etapa: Esta fase inicial estará dedicada a mi formación en Unreal Engine 4 (UE4), el motor seleccionado para el desarrollo del proyecto, así como en el uso de *Blueprints* y el estudio del sistema de agarre del proyecto UnrealTrackingGrasp del que se partirá.
- Segunda etapa: Durante este tiempo, realizaré un estudio de mercado y analizaré la problemática específica para idear la base del *escape room*. Posteriormente, comenzaré la programación de la base del proyecto, que incluirá el desarrollo que incluirá el desarrollo de un sistema de puzzles y otro para seguimiento del jugador.
- Tercera etapa: Durante esta etapa me dedicaré al diseño definitivo de los puzzles, los niveles y el entorno. Esta etapa constará de una subetapa dedicada al modelado de todos los elementos necesarios según los puzzles diseñados.
- Cuarta etapa: La fase final será una de pulido de detalles para asegurar la calidad y coherencia del proyecto con la propuesta asignada.
- Paralelamente al desarrollo del proyecto, continuaré redactando este documento como memoria del trabajo de fin de grado.

2 Estado del Arte

Este capítulo, en el que se revisan conceptos básicos y trabajos relacionados con la propuesta, se divide en varias secciones: la sección 2.1, donde se plantea un estudio de la problemática relacionada con el deterioro cognitivo; la sección 2.2, la cual profundiza en el deterioro cognitivo y estrategias preventivas; la sección 2.3, en la que se explora la gamificación como herramienta para frenar el deterioro cognitivo; la sección 2.4, aborda la realidad virtual y su evolución hasta la fecha; y finalmente, la sección 2.5, se centra en los *escape rooms*, revisando su historia y evolución.

2.1 Estudio de la problemática

El deterioro cognitivo es una seria preocupación para las sociedades más avanzadas. España no es ninguna excepción, con el aumento de la población adulta mayor y con una esperanza de vida que constantemente se alarga, el crecimiento de personas que sufren de deterioro cognitivo sigue el mismo camino. Un estudio publicado en 2018 [26] muestra que la edad es un factor de riesgo para el desarrollo de Deterioro Cognitivo Leve (DLC); los datos indican que a medida que las personas se hacen mayores, aumenta la probabilidad de que sufran deterioro cognitivo leve. El estudio también sugiere que las personas con DLC tienen un riesgo significativo de progresar a demencia, se observó que aproximadamente el 15% de los sujetos mayores de 65 años con DLC desarrollaron demencia tras un seguimiento de 2 años. Datos preocupantes al conocer qué según estudios recientes, España se coloca como el tercer país con mayor esperanza de vida mundial¹.



(a) Mapa de color del mundo según esperanza de vida al nacer

Comparativa: Esperanza de vida al nacer					
Países	Fecha	Esperanza de vida - Mujeres	Esperanza de vida - Hombres	Esperanza de vida	Var.
San Marino [+]	2012	86,80	84,10 85,60	84,10 85,60	2,73%
Liechtenstein [+]	2023	87,30	82,50 84,80	82,50 84,80	1,07%
Suiza [+]	2023	85,90	82,30 84,20	82,30 84,20	0,60%
España [+]	2023	86,70	81,20 84,00	81,20 84,00	1,11%
Japón [+]	2022	87,09	81,05 84,00	81,05 84,00	-0,53%
Italia [+]	2023	85,80	81,70 83,80	81,70 83,80	1,21%
Hong Kong [+]	2022	86,77	80,70 83,66	80,70 83,66	-2,19%
Mallorca [+]	2023	85,30	81,80 83,60	81,80 83,60	1,46%
Andorra [+]	2022	85,70	81,10 83,50	81,10 83,50	-0,24%
Luxemburgo [+]	2023	85,10	81,70 83,40	81,70 83,40	0,48%
Suecia [+]	2023	85,00	81,70 83,40	81,70 83,40	0,36%
Australia [+]	2022	85,30	81,20 83,20	81,20 83,20	-0,12%
Francia [+]	2023	85,90	80,10 83,10	80,10 83,10	0,97%

(b) Tabla de los países con mayor esperanza de vida, general y por géneros

Figura 2.1: Esperanza de vida global por países en 2023

Todos estos datos ya se ven reflejados en la población, donde, por ejemplo, en Reino Unido,

¹<https://datosmacro.expansion.com/demografia/esperanza-vida>

un estudio publicado en la revista *Clinical Epidemiology* concluyó que las consultas médicas vinculadas con el deterioro cognitivo se han triplicado en los últimos 10 años. Las perspectivas futuras[22] sugieren un panorama realmente negativo; se prevé que el número de adultos con 40 años o más con demencia llegue a casi triplicarse para el año 2050, cuando se alcanzará una cifra de 153 millones de casos a nivel mundial, en comparación a los 57 millones de casos en 2019. El estudio aporta datos específicos por países, en él se puede observar que en Europa Occidental la demencia aumentará de casi ocho millones en 2019 a casi 14 millones en 2050. Se estiman incrementos relativamente pequeños de casos, por ejemplo, en Grecia del 45%, en Italia del 56% o en Finlandia del 58%. En España, sin embargo, se espera un incremento del 83% del número de casos de personas con demencia, llegando hasta más de un millón y medio en 2050 únicamente en el territorio nacional.

Ante este sombrío panorama, la necesidad de encontrar métodos innovadores y efectivos para la prevención del deterioro cognitivo es más urgente que nunca. En este contexto, la creación de un proyecto de *escape room* temático surge como solución ante la búsqueda. Este proyecto está diseñado como una forma de terapia gamificada con el objetivo de estimular las diversas capacidades cognitivas de las personas mayores participantes. Para entrenar estas habilidades se integran desafíos y puzzles que requieren el uso de memoria, lógica, lenguaje, razonamiento y habilidades resolutivas. El *Scape Room* proporciona una estimulación mental que podría ayudar a retrasar o incluso prevenir la progresión del deterioro cognitivo, consiguiendo así alinearse con las necesidades de una población cada vez más longeva.

2.2 Deterioro Cognitivo

Somos humanos, nacemos, crecemos y envejecemos; en todas estas etapas hay cambios y, cuando envejeces, no es una excepción: surgen nuevas dificultades, en nuestra agilidad o en la precisión de nuestras condiciones cognitivas. Sin embargo, hay cambios cognitivos que no se justifican con la edad, sino que son consecuencia de alguna condición médica. Es a esto a lo que se denomina deterioro cognitivo. Este trastorno puede afectar a una o varias capacidades cognitivas, como pueden ser la memoria, el lenguaje, la atención o la capacidad de razonar, y con ello, a la calidad de vida de los pacientes.

Según el grado de gravedad, existen diferentes tipos de deterioros cognitivos:

1. DLC: Conocido también por sus siglas en inglés como Mild Cognitive Impairment (MCI), representa una situación de pérdidas de memoria o alteraciones relacionadas con la cognición, pero estas no afectan al funcionamiento cotidiano sin repercutir de forma significativa en la vida del paciente. El DLC constituye una fase intermedia entre el envejecimiento normal y la demencia leve producida por alguna enfermedad cerebral, como el Alzheimer. A su vez, el DLC se puede clasificar en diferentes tipos, siendo los dos principales el amnésico y no amnésico. El DLC amnésico está caracterizado por déficits de memoria más pronunciados que los asociados al envejecimiento sano. La persona sigue siendo independiente y no experimenta ni confusión ni desorientación. Por otra parte, el DLC no amnésico se caracteriza por afectar a capacidades cognitivas fuera de la memoria, como podría ser el lenguaje, la atención, la planificación y la ejecución de tareas complejas.

2. Deterioro Cognitivo Medio o Demencia Leve: En este grado, el deterioro comienza a afectar la situación funcional del paciente; comienzan a resentirse de forma leve sus actividades diarias. Este deterioro se caracteriza por presentar cierto grado de desorientación espacial y temporal. Asimismo, se evidencian problemas con la memoria reciente, además de presentar cambios en la conducta y estado de ánimo, como una mayor irritabilidad.
3. Deterioro Cognitivo Notable o Demencia Moderada: En este caso, los síntomas empiezan a ser más evidentes; el paciente presenta dificultades en el aprendizaje, a la hora de comprender y ejecutar órdenes. Es frecuente que el trastorno afecte al comportamiento y conducta del paciente, generando agresividad, estados de ansiedad o llegando a causar depresión.
4. Deterioro Cognitivo Severo o Demencia Grave: Llegados a este grado de deterioro, se hace imposible comunicarse con el paciente; él no comprende el lenguaje y tampoco tiene sentido lo que dice. Afecta con una pérdida total de la memoria, ya no solo a la inmediata, sino también a la de largo plazo. Convierte al paciente en completamente dependiente.

2.2.1 Estrategias Preventivas y Fomento de la Reserva Cognitiva

Actualmente, las investigaciones se centran especialmente en un enfoque preventivo [24], cuyo objetivo es retrasar lo más posible la aparición de los primeros síntomas. Participar en actividades cognitivamente estimulantes como la lectura o los juegos mentales es una estrategia que puede desempeñar un papel crucial en la reducción del riesgo de deterioro cognitivo leve y en la prevención de su progresión a demencia. Otro buen aliado es contar con una alta reserva cognitiva. La acumulación de experiencia educativa y la estimulación de las capacidades mentales a lo largo de la vida se refleja en lo que llamamos *reserva cognitiva*. Cuanto mayor sea esta, más nos ayudará en la eficiencia de nuestras capacidades cognitivas, es un factor que contribuye a retrasar el posible deterioro cognitivo, promoviendo una red neuronal más resistente y compensando los efectos del envejecimiento o enfermedades neurodegenerativas.

A pesar de no existir ejercicios específicos que eviten sufrir de DLC o demencia, sí que existen una serie de prácticas muy recomendables para favorecer la reserva cognitiva [19]. Según se ha estudiado, es importante tener una mente abierta a nuevas actividades que salgan de nuestra zona de confort y nos planteen retos. Puesto que el cerebro y sus capacidades cognitivas funcionan como una red, es importante entender que hay que trabajar todas las capacidades de modo conjunto; por ejemplo, para mejorar la memoria o el lenguaje es importante trabajar también la atención y el razonamiento. Comprendiendo esto, se entiende que más que dedicarle mucho tiempo es más importante la variedad de ejercicios que estimulen distintos campos y resulten motivadores.

Dentro de las actividades recomendadas para aumentar la reserva cognitiva encontramos jugar y ponerse a prueba. Jugar permite entrenar distintas habilidades cognitivas según el tipo de juego, como podrían ser el cálculo, la memoria reciente, la lógica, la capacidad de planificación, el vocabulario o la creatividad. De la misma forma, es importante ponerse

a prueba de forma diaria, como podría ser resolviendo puzzles. Es ahora cuando conviene presentar el concepto de la gamificación.

2.3 Gamificación

El término gamificación es un neologismo de la palabra inglesa *gamification*, originada del inglés *game*. Es frecuente encontrar la palabra *ludificación* como equivalente hispanizado. El término *Gamification* fue acuñado por primera vez por Nick Pelling 2002, refiriéndose a la aplicación de elementos y diseños de los juegos en contextos fuera del entretenimiento. Para entenderlo mejor, se citará a los profesores Gallego, Molina y Llorens en las XX Jornadas sobre la Enseñanza Universitaria de la Informática desde el departamento de Ciencia de la Computación e Inteligencia Artificial de Alicante: '*Gamificación (o ludificación) es el uso de estrategias, modelos, dinámicas, mecánicas y elementos propios de los juegos en contextos ajenos a estos, con el propósito de transmitir un mensaje o unos contenidos o de cambiar un comportamiento, a través de una experiencia lúdica que propicie la motivación, la implicación y la diversión*' [11] .

Actualmente, las estrategias terapéuticas están basadas en completar de forma regular y monótona diversas acciones una tras otra, convirtiéndose en una rutina para los pacientes, lo que suele llevar a aburrimiento por parte de los pacientes y a su vez a su abandono en un corto período de tiempo. La gamificación busca solucionar este problema convirtiéndose en una técnica eficaz en diversos ámbitos como el de la salud y el bienestar con un objetivo motivacional. Uno de nuestros objetivos como desarrolladores es que al usuario nuestro proyecto le resulte entretenido: '*Gamificar es plantear un proceso de cualquier índole como si fuera un juego. Los participantes son jugadores y como tales son el centro del juego, y deben sentirse involucrados, tomar sus propias decisiones, sentir que progresan, asumir nuevos retos, participar en un entorno social, ser reconocidos por sus logros y recibir retroalimentación inmediata. En definitiva, deben divertirse mientras se consiguen los objetivos propios del proceso gamificado*' [11] .

La gamificación tiene beneficios en muchos ámbitos, entre ellos el cuidado de personas mayores que presentan deterioro cognitivo. Entre los beneficios encontramos:

- Permite mantener o incluso mejorar la salud mental, física y emocional de las personas mayores proporcionándoles diversas actividades estimulantes. La gamificación mejora el bienestar y la calidad de vida de los pacientes. El uso por parte de las personas mayores de una aplicación de entrenamiento cognitivo puede ayudar a sus capacidades cognitivas como la memoria, atención o lenguaje a través de presentarles tareas que les supongan un reto.
- La gamificación puede aumentar el sentimiento de autonomía y empoderamiento de las personas mayores, ayudándoles a sentir más control sobre sus propias vidas al presentarles objetivos y retroalimentación.

La gamificación tiene la capacidad de mejorar o mantener la función cognitiva y prevenir el deterioro cognitivo entre los adultos mayores, a través de distintos medios, como:

- Ofrecer entrenamiento y estimulación cognitiva para que las personas mayores puedan ejercitarse su mente y poner a prueba sus habilidades mentales, recibiendo a su vez retroalimentación y recompensas.
- Incentivar el aprendizaje y despertar la curiosidad, permitiendo a los adultos mayores la capacidad de adquirir nuevos conocimientos y habilidades, explorando a su vez nuevos temas y ámbitos.
- Impulsar la creatividad y la innovación, donde los pacientes puedan expresarse libremente y generar nuevas ideas.
- Desarrollar nuevas experiencias inmersivas y que resulten atractivas a las personas mayores para que puedan disfrutar explorando entornos virtuales, interactuando con personajes y objetos virtuales.

A pesar de sus beneficios, la gamificación también plantea algunas cuestiones éticas, técnicas y prácticas que es importante abordar:

1. Es importante estudiar y comprender las necesidades de las personas mayores para conseguir adaptar exitosamente la gamificación a sus necesidades y preferencias. De igual forma es crucial entender que no todos los adultos mayores son iguales y pueden tener diferentes objetivos, intereses, capacidades y limitaciones. Además, la gamificación debe considerar las diferencias culturales y generacionales. Por ejemplo, no sería lo mismo diseñar un juego para una persona mayor japonesa que para una española, podría no funcionar ni ser adecuado.
2. Se debe equilibrar los beneficios frente a los riesgos de la gamificación. Es cierto que la gamificación puede tener efectos positivos en las personas mayores que ya hemos mencionado anteriormente. Sin embargo, también puede acarrear efectos negativos, como provocar adicción, frustración y aburrimiento. Por lo tanto, la gamificación debe diseñarse buscando que se maximicen los beneficios y se minimicen los riesgos para los adultos mayores brindándoles retroalimentación, orientación y apoyo.

En los últimos años, la gamificación cada vez se hace notar más y tiene mayor repercusión como técnica en ámbitos como el del entretenimiento y la educación principalmente. Esto viene a raíz del hecho de que ha crecido a la par de las nuevas tecnologías como la Realidad Aumentada o la Realidad Virtual.

2.4 Realidad Virtual

Existen multitud de definiciones acerca de qué es la Realidad Virtual (RV) , una palabra que se repite continuamente es simulación. Se puede decir que la RV es la tecnología que permite simular un entorno y que ofrece al usuario una experiencia inmersiva en un mundo virtual. El objetivo de esta experiencia es engañar los sentidos del usuario, buscando hacerle sentir la sensación de estar dentro de otro mundo. Actualmente, los sistemas estándar de realidad virtual logran simular la presencia física del usuario dentro de un entorno virtual utilizando, en la mayoría de los casos, gafas de RV equipadas con sonido 3D y sensores de movimiento y

controladores. Gracias a estos elementos, una persona que utiliza equipos de realidad virtual puede observar el mundo artificial, moverse dentro de él e interactuar con sus elementos.

Todo sistema de RV debe contar con tres elementos básicos para conseguir crear un entorno virtual realista [5]:

- **Simulación interactiva:** La interacción con elementos del nuevo mundo creado y su realismo son clave a la hora de conseguir la sensación de inmersión. A mayor interacción pueda tener el usuario con el entorno virtual o mejor sea la acción/reacción por parte del mundo virtual, más realista será la experiencia para el usuario.
- **Interacción implícita:** La interacción implícita es clave en la RV, está definida por los gestos naturales del usuario, como la rotación de la cabeza o el movimiento de las manos, definen las interacciones en el mundo virtual todas aquellas acciones basadas los movimientos naturales del usuario, como los giros de cabeza o movimientos de la manos, esta será la forma que tendrá el usuario de comunicar su intencionalidad, eliminando la necesidad de utilizar órdenes , o dispositivos clásicos como teclados y ratones. Tecnologías como el *hand tracking*, el seguimiento ocular o el uso de giroscopios en las gafas de RV permiten un control fluido y natural de la cámara y de los objetos en la escena. Todas estas tecnologías permiten mejorar la experiencia inmersiva, haciendo que el usuario sienta que se encuentra dentro de otro mundo y no que esta interactuando con una máquina.
- **Inmersión sensorial:** La inmersión sensorial es la habilidad de la tecnología para generar una desconexión de los sentidos de la realidad, creando una vivencia multisensorial que brinda a los usuarios una sensación intensa y profunda de estar inmersos en un entorno virtual. A pesar de ser conscientes de que es una simulación por computadora, la experiencia logra que sientan una presencia real en dicho mundo virtual. Esta desconexión de los sentidos del mundo real se logra mediante estímulos sintéticos que reemplazan a los estímulos reales, como gráficos de alta calidad, sonido 3D envolvente y dispositivos de visualización avanzados. La visión estereoscópica desempeña un papel crucial, proporcionando profundidad y realismo al entorno virtual. Al integrar todos estos elementos, el entorno simulado se siente más auténtico, mejorando significativamente la experiencia del usuario.

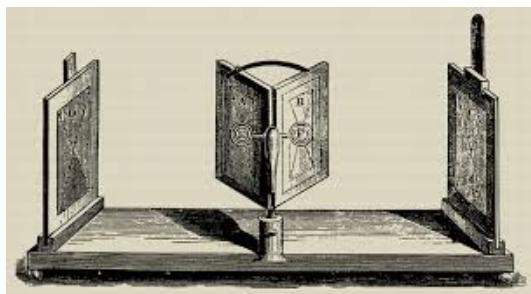
2.4.1 Evolución de la Realidad Virtual

En 1987, Jaron Lanier acuñó el término “realidad virtual”[4] durante un periodo de intensa investigación en tecnologías inmersivas. Junto con Thomas Zimmerman, fundó VPL Research, la primera empresa en comercializar equipos de realidad virtual, como el *DataGlove*, el *EyePhone HMD* y la *Audio Sphere*. Su empresa fue la responsable de producir los primeros equipos de realidad virtual disponibles para uso comercial accesibles al público, contaban con unas gafas de realidad virtual y guantes de datos. Su trabajo popularizó el concepto y sentó las bases para el desarrollo de tecnologías inmersivas.

2.4.1.1 Orígenes históricos de la Realidad Virtual

Aunque el término realidad virtual se consolidó en los años 80, sus raíces pueden rastrearse hasta 1793, cuando el pintor irlandés Robert Barker desarrolló una tecnología inmersiva exhibiendo unas pinturas panorámicas. Para ello, patentó una tecnología inmersiva que consistía en un edificio construido específicamente para mostrar pinturas ubicadas en la cara interna de paredes circulares.

En 1838, Charles Wheatstone, a partir de un estudio donde demostraba que el cerebro combina dos fotografías del mismo objeto para conseguir una sensación de profundidad e inmersión, inventó el estereoscopio (Figura 2.2a), un dispositivo que utilizaba dos espejos para combinar imágenes y generar una sensación de profundidad. Este invento marcó un hito en la percepción visual inmersiva, siendo utilizado durante más de un siglo en entretenimiento, educación e incluso juguetes infantiles (Figura 2.2b).



(a) *The Wheatstone mirror stereoscope*



(b) Estereoscopio infantil

Figura 2.2: Invento del estereoscopio y su uso posterior

El cine inmersivo comenzó en 1922 con *The Power of Love*, la primera película en 3D, que empleaba el método de anaglifo y requería gafas de colores complementarios. Décadas más tarde, Morton Heilig revolucionó esta experiencia con su *Sensorama*, un dispositivo multisensorial que combinaba imágenes 3D con sonido estéreo, viento, olores y vibraciones, aunque no tuvo éxito comercial. No sería hasta décadas más tarde que la realidad virtual volvería al mundo del entretenimiento.

2.4.1.2 Avances tecnológicos en el siglo XX

En los años 80, la realidad virtual comenzó a consolidarse como una tecnología con aplicaciones más diversas. En 1982, Sandin y DeFanti desarrollaron los *Sayre Gloves*, los primeros guantes de datos con sensores ópticos para rastrear movimientos de los dedos, una innovación clave en el reconocimiento de gestos.

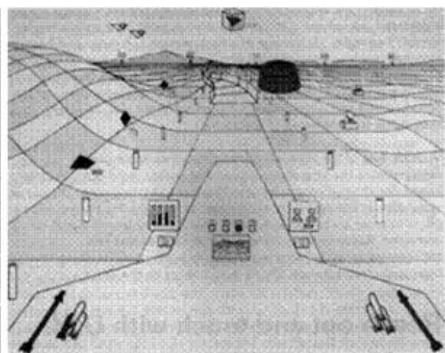
En el ámbito militar, Thomas Furness diseñó el simulador VCASS y, más tarde, el Super Cockpit (1986-1989) (Figura 2.3a), un sistema que permitía controlar aviones mediante gestos, voz y movimientos oculares. Este enfoque integró datos computarizados avanzados

y sistemas de seguimiento, revolucionando la formación de pilotos. En 1997 se desarrolló el proyecto “Virtual Vietnam” donde se utilizaba la RV para crear escenarios de zonas de guerra destinados a usarse en terapia de exposición para veteranos con trastorno de estrés postraumático.

En 1989, Scott Foster fundó Crystal River Engineering y, bajo contrato de la NASA, desarrolló el proyecto *VIEW* (Virtual Environment Workstation Project) (Figura 2.3b), que introdujo procesamiento de audio binaural 3D en simuladores de entrenamiento. Ese mismo año, Mattel lanzó el *Power Glove*, un accesorio para Nintendo basado en el *DataGlove*, aunque tuvo una recepción comercial limitada.



(a) Sistema Super Cockpit



(b) Proyecto VIEW

Figura 2.3: Avances en el siglo XX en la tecnología de la realidad virtual

2.4.1.3 Auge de la Realidad Virtual en el entretenimiento

En los 90, surgieron sistemas como *Virtuality*, una serie de máquinas *arcade* que ofrecían experiencias inmersivas en 3D y soporte para juegos multijugador. Paralelamente, SEGA desarrolló el *SEGA VR* (Figura 2.4a) y, aunque nunca llegó al mercado, fue precursor de los visores modernos. Nintendo lanzó la consola *Virtual Boy* en 1995 (Figura 2.4b), pionera en gráficos 3D portátiles, aunque fracasó debido a su diseño incómodo y falta de soporte.

2.4.1.4 Renacimiento de la Realidad Virtual

En 2010, se presentó el prototipo del Oculus Rift, que ofrecía un campo de visión de 90 grados. Este dispositivo revitalizó el interés en la realidad virtual, y en 2012 una exitosa campaña de Kickstarter para Oculus Rift recaudó 2,4 millones de dólares. En 2014, Facebook adquirió Oculus por 2.000 millones de dólares, marcando un punto de inflexión en la industria.

En 2015, empresas como Google impulsaron el acceso masivo con dispositivos asequibles como *Google Cardboard*. Para 2016, cientos de empresas estaban desarrollando productos de realidad virtual. HTC lanzó su *headset HTC VIVE SteamVR*. Fue el primer lanzamiento



Figura 2.4: Realidad virtual en el mundo del entretenimiento

comercial de un casco con seguimiento basado en sensores, lo que permitía a los usuarios moverse libremente en un espacio. Desde entonces, muchas empresas se encuentran desarrollando sus propios *headsets* de RV incluyendo a Apple, Amazon, Microsoft, Sony, Samsung, etc. Más recientemente, en 2023, Apple anunció su visor de realidad mixta Apple Vision Pro, y Meta presentó el Quest 3, ambos con innovaciones en resolución, seguimiento y capacidades de realidad mixta.

2.4.1.5 Aplicaciones contemporáneas

Esta tecnología, cada vez más utilizada, destaca por su versatilidad, se le han encontrado en los últimos años una gran variedad de aplicaciones. Gracias a los constantes avances, el desarrollo y uso de entornos virtuales se puede encontrar cada vez en más sectores como puede ser: como la medicina, donde ya se utiliza en ámbitos educativos, recreativos y terapéuticos. Uno de sus mayores usos es la formación, donde se emplea para entrenar desde cirujanos, pilotos, cuerpos de emergencia y seguridad hasta profesionales de todos los campos. Sin embargo, el sector donde esta tecnología más ha triunfado y más se ha desarrollado es en el del entretenimiento. A raíz de la adaptación de Steven Spielberg de la novela *Ready Player One*² esta tecnología se ha viralizado a nivel mundial, se han creado videojuegos de todos los géneros desde terror hasta aventura pasando por los *Escape Rooms*.

2.5 Escape Room

El término *escape room* proviene del inglés, su traducción al castellano es juego de escape, pero también se le conoce en inglés como *escape game*, *puzzle room*, *exit game* o *riddle room*. Un *escape room* es un subgénero de *point-and-click*, algo que más adelante veremos a qué se refiere, donde el objetivo del jugador es explorar sus alrededores para poder escapar de su encarcelamiento. La sala o habitación usualmente consistía en una puerta cerrada, objetos para manipular y pistas o compartimentos secretos. Mediante la interacción con todos estos

²[https://en.wikipedia.org/wiki/Ready_Player_One_\(film\)](https://en.wikipedia.org/wiki/Ready_Player_One_(film))

elementos, el jugador debía encontrar la forma de escapar.

Los *escape rooms* ponen a prueba la capacidad de resolver problemas y el pensamiento lateral de los participantes al proveer varios y diversos puzzles y retos en el juego. Entre los tipos de puzzles comúnmente se pueden encontrar juegos de palabras, numéricos, puzzles de secuencias o patrones, entre otros.

2.5.1 Historia del Escape Room

El origen de los *escape rooms* se encuentra en el ámbito de los videojuegos. En los primeros tiempos de la industria de los videojuegos, el ordenador personal medio no contaba con la tecnología necesaria para poder producir imágenes de calidad, aún menos imágenes que debían transportarte a un mundo de imaginación. La solución fueron los juegos basados en comandos de texto. Estos juegos usaban únicamente el texto para transportar al jugador a través de mundos y realidades. Esto permitía a los desarrolladores superar las limitaciones del *hardware*. En este género de texto de aventura se pueden encontrar grandes títulos como *Zork* [15], *The Hitchhiker's Guide to the Galaxy*[16] y *Colossal Cave Adventure*[7].

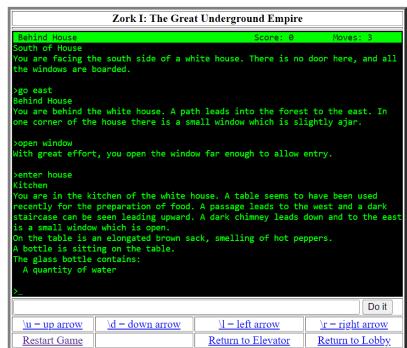
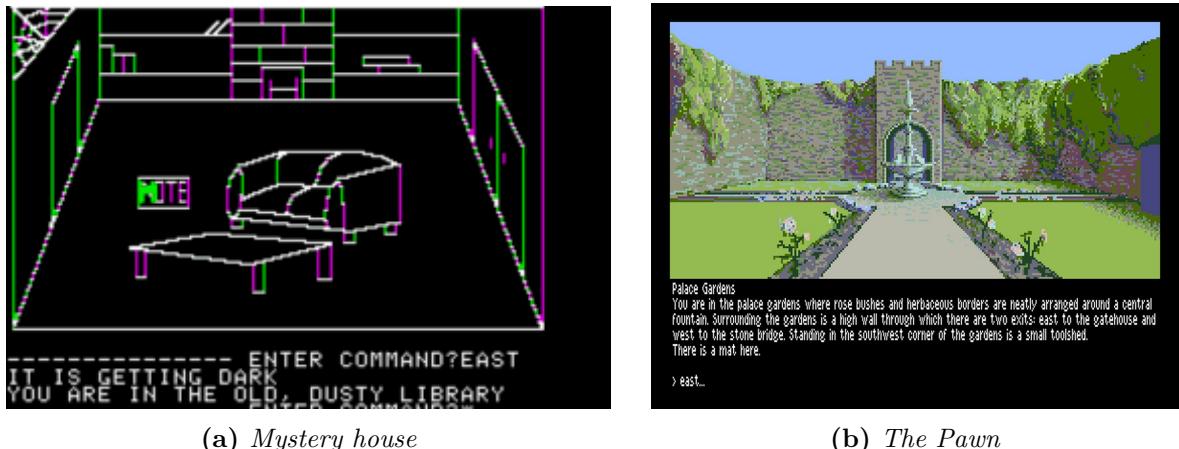


Figura 2.5: Captura del juego Zork

Era bien sabido que este género se beneficiaría en gran medida de los elementos visuales en pantalla para acompañar a los textos descriptivos. Este cambio sucedió a inicios de la década de 1980 con títulos como *The Pawn* [27] o *Mystery House*[32] que fusionaban imágenes rudimentarias junto a palabras para mejorar la experiencia inmersiva.

(a) *Mystery house*(b) *The Pawn*Figura 2.6: Capturas de los juegos *Mystery house* y *The Pawn*

Conforme la tecnología fue avanzando también lo hicieron los videojuegos dando paso a los juegos de aventura *point-and-click* considerados los padres de los *escape rooms*. Estos juegos se basaban en un método donde se usaba el cursor del ratón para hacer clic por la pantalla lo que permitía así moverse por ella o interactuar con sus elementos. Un ejemplo sería *Planet Mephius*[34] (1983).

Figura 2.7: Captura del juego *Planet Mephius*

Más adelante, en 1988, John Wilson desarrolló *Behind Closed Doors* [28], un juego basado en texto que se trata del primer juego con el jugador atrapado dentro de una habitación, en este caso un baño.



(a) Portada del juego



(b) Ejemplo de jugabilidad

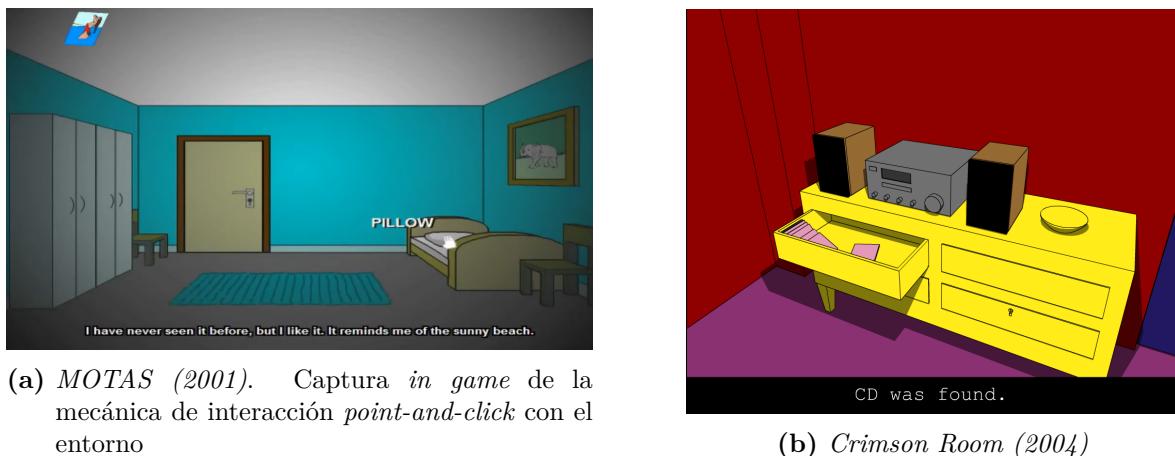
Figura 2.8: Elementos del videojuego *Behind Closed Doors*

El concepto del juego ganó terreno a lo largo de la siguiente década y media generando que los juegos de escape adoptasen el estilo *point-and-click*. Sin embargo, el aumento de juegos de escape se atribuye a la popularidad de la saga *Myst* [8] (1993) en el cual los jugadores tenían que buscar pistas por el entorno.

El término *escape room* fue acuñado en 2001 a partir de *Mystery of Time and Space*[2], también conocido como “MOTAS”, una aventura gráfica creada por Jan Albartus que fue producida usando Macromedia Flash (*Adobe Flash*³ en la actualidad). Este juego también empleó el método *point-and-click* para buscar por el entorno distintos elementos escondidos; luego, estos objetos recuperados podían usarse para interactuar de nuevo con el entorno u otros elementos.

Pocos años más tarde, en 2004 se publicó *Crimson Room*, considerado uno de los juegos más influyentes en el género *escape the room*. otra muestra junto al ya mencionado MOTAS de cómo los juegos ya no dependían únicamente del género de aventura que había sido tan influyente en las décadas predecesoras.

³<https://www.adobe.com/es/products/flashplayer/end-of-life-alternative.html>



(a) MOTAS (2001). Captura *in game* de la mecánica de interacción *point-and-click* con el entorno

(b) Crimson Room (2004)

Figura 2.9: MOTAS (2001)

Hoy en día, numerosos negocios escogen el camino tecnológico, pasando de la vida real al mundo digital, sin embargo, Takao Kato no siguió esta corriente cuando en 2007 abrió su primer *escape room* en la vida real, lo llamó “The Real Escape Game”. Kato fue inspirado por la popularidad de los videojuegos *escape the room* y por su deseo de vivir aventuras como las que sucedían en los libros que leía. *‘Me preguntaba por qué no pasaban cosas interesantes en mi vida como las que pasan en los libros. Pensé que podía crear mi propia aventura -una historia- y, entonces, invitar a personas a formar parte de ella... Una chica sentada a mi lado me contó de su obsesión con los juegos de escape online, así que simplemente traté de hacer uno’* - Takao.



Figura 2.10: Takao Kato posando dentro de “The Real Escape Game”

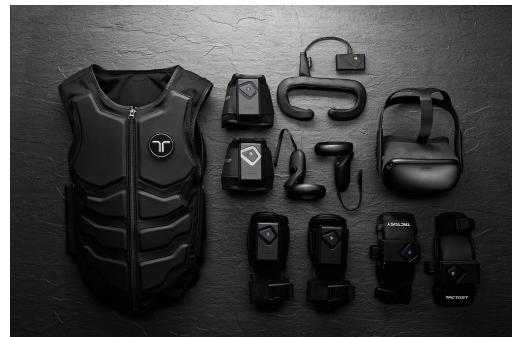
La población buscaba nuevas formas de entretenerte, ya no querían sentarse y observar, querían sentirse partícipes, querían hacer cosas. En este contexto, a finales de 2014 la industria vio un estallido de los negocios de juegos de escape en la vida real. Hoy en día se pueden encontrar una aventura *escape room* en todas las grandes ciudades alrededor del mundo, y se estima que a fecha de noviembre de 2019 había globalmente entre 50,000 y 60,000 juegos de escape.

Más allá, los *escape room* han dado un paso más en su evolución fusionándose con la realidad virtual pasándose a llamar “VR Escape Room”. No es difícil de entender por qué una experiencia inmersiva como los *escape room* van tan bien en conjunto de la realidad virtual con su inmensa capacidad de inmersión del usuario. En los *VR Escape Rooms*, los jugadores cuentan con un *headset* que permite transportarlos a un entorno virtual donde pueden interactuar con sus alrededores usualmente con la ayuda de unos controladores para resolver puzles y progresar en el juego. Esta tecnología pone al alcance de nuestras manos ilimitadas posibilidades y aventuras.

La última novedad dentro de los *VR Escape Room* son las experiencias multisensoriales, son pocos por ahora pero ya existen instalaciones de juegos de escape que incorporan trajes hápticos (Figura 2.11b) y pistolas para que el jugador pueda sentir las vibraciones del juego y obtener un mayor *feedback* de sus acciones. Un traje háptico es aquel que está equipado con sensores internos y externos que se conectan con los nervios del usuario para generar diversas sensaciones táctiles. Un ejemplo de estas experiencias multisensoriales se encuentra en Londres gracias a la compañía *OMESCAPEVR*⁴.



(a) Experiencia *VR Escape Room*



(b) Traje para experiencia multisensorial

Figura 2.11: algo aquí de los juegos

En este proyecto vamos a volver a los orígenes de los *escape rooms* creando un videojuego. Se empleará la tecnología desarrollada previamente por el laboratorio 3DPLab de *handtracking* para VR (seguimiento de manos)⁵ eliminando de esta forma los controladores de las gafas y creando una interacción mucho más natural.

2.6 Antecedentes

En el proyecto se propone la creación de un *escape room* para Realidad Virtual (RV), utilizando tecnología de seguimiento manual (*hand tracking*) con el fin de prevenir el deterioro cognitivo. Para contextualizar, clasificaré las alternativas existentes en el mercado en dos categorías: una centrada en la tecnología de *escape rooms* en RV con *hand tracking*, y otra enfocada en *escape rooms* y actividades diseñadas específicamente para personas mayores.

⁴<https://omescapelondon.co.uk/>

⁵<https://github.com/3dperceptionlab/unrealtrackinggrasp>

Actualmente, hay diversos juegos de *escape room* para realidad virtual disponibles, como *Escape Simulator* [30] lanzado el 19 de octubre de 2021, que se caracteriza por ofrecer una experiencia de inmersión donde los jugadores resuelven acertijos y puzzles en entornos con una temática específica.



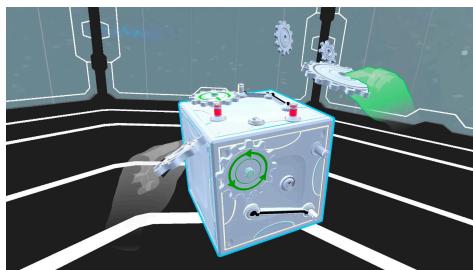
(a) Cartel del juego *Escape Simulator*



(b) Captura de un nivel de *Escape Simulator*

Figura 2.12: Antecedente de un *escape room* para RV *Escape Simulator*[30]

Sin embargo, al añadir el componente de *hand tracking*, las opciones se reducen significativamente al ser una tecnología todavía novedosa. Hay pocos ejemplos de juegos que empleen esta tecnología, y aún menos que estén enfocados en temáticas de *escape room*. Un juego que emplea un sistema de seguimiento de manos avanzado es *Mindset* [29], un juego de rompecabezas en cubo que supone un reto de desafíos intelectuales. Que cumpla el resto de requisitos únicamente está disponible *Escape Fear*, una *demo* de *escape room* de horror con *hand tracking* para Meta Quest 3 que dura solo unos minutos. Más allá, existen negocios en algunas ciudades que ofrecen actividades incluyendo *escape rooms* con tecnología de *hand tracking*, pero su accesibilidad es limitada por su ubicación, lo cual no es ideal para potenciales residentes.



(a) Captura en juego de *Mindset*



(b) Captura del juego *Escape Fear*

Figura 2.13: Antecedentes de juegos con puzzle y seguimiento de manos

Por otro lado, hay actividades y experiencias educativas[12] de tipo *escape room* muy completas que se han implementado en residencias de ancianos con objetivos similares a este proyecto. Estas actividades prescinden de la parte digital pero se benefician de todas las ventajas de la gamificación. Asimismo, existe un estudio en el que a un grupo de ancianos se les presentó un juego de RV, recibieron su *feedback* y desarrollaron la idea de un *escape room*

digital para personas mayores, aunque finalmente no se llevó a cabo[9]. Este estudio reveló aspectos importantes como la necesidad de mejor iluminación, puzzles más variados, tiempos más flexibles, y controles menos confusos.



(a) Residente resolviendo puzzles



(b) Presentación de la dinámica a los residentes

Figura 2.14: Experiencia *escape room* en residencia en Ciempozuelos

Por último, mencionar cómo en el proyecto del que partimos⁶ se llegó a realizar una demo técnica con la intención de evaluar la efectividad del sistema de agarre en situaciones reales. El resultado fue positivo, demostrando que el sistema es funcional y que posee el potencial necesario para ser implementado en el desarrollo de un *escape room* más completo.

⁶<https://github.com/3dperceptionlab/unrealtrackinggrasp>

3 Metodología

Este capítulo, en el que se describen las diversas herramientas y dispositivos que se usarán para desarrollar este proyecto, se divide en tres secciones: la sección 3.1, en la que se explica todo lo necesario para realizar el proyecto; la sección 3.2, en la que se detallan los dispositivos utilizados para el desarrollo; y por último, la sección 3.3, en la que se presentan las herramientas usadas a lo largo de todo el proyecto.

3.1 Introducción

Se ha decidido usar UE4 como el software principal a la hora de desarrollar el proyecto por su capacidad de adaptarse a las necesidades específicas de los desarrolladores, y cuenta con todo lo necesario para desarrollar nuestra aplicación. Destaca también por contar con un gran entorno gráfico del que en este proyecto podrá sacar provecho a la hora de realizar los entornos virtuales. Otra ventaja del motor es su capacidad de proporcionar resultados realistas en tiempo real, reduciendo mucho el tiempo que se habría dedicado originalmente al renderizado. Para el momento en el que se empezó el desarrollo del trabajo ya se había lanzado Unreal Engine 5 (UE5), una versión más potente y actualizada del motor de Epic Games. Sin embargo, hay que tener en cuenta el hecho de que partimos de un proyecto anterior donde se desarrolló un seguimiento de manos empleando UE4. El paso del trabajo a esta nueva versión del motor conllevaría unos costos excesivamente altos de tiempo que no se justifican. Más allá, en el paso de una versión a otra podrían surgir complicaciones o incompatibilidades que podrían resultar desastrosas para el proyecto.

Con respecto al *hardware* utilizado, la decisión final ha sido mantener las gafas seleccionadas en el proyecto del que se parte, en este caso se refiere a las Meta Quest, que cuentan con muchas ventajas desde la compatibilidad con el motor escogido hasta la infinidad de documentación tanto por parte de Unreal Engine (UE), de Meta o disponible en foros relacionados. Además, surge el dilema de compatibilidad y del costo de programación para adaptarlo a unas nuevas gafas. Actualmente, la empresa Meta cuenta con tres modelos de gafas, el último de los cuales cuenta con una gran diferencia con respecto a las anteriores: la realidad mixta. A pesar de esto, los dos primeros modelos ya aportan con todo lo necesario para el proyecto por lo que serán seleccionadas para el proyecto.

3.2 Hardware

En esta sección trataremos tanto las gafas de realidad virtual empleadas en el proyecto como el ordenador utilizado para su desarrollo. Es importante que ambos cumplan ciertos criterios, especificaciones y mínimos, sobre los cuales nos expliqueremos en la sección 3.2.4. Las gafas Meta Quest han sido realmente beneficiosas, ofreciendo grandes resultados en el sistema de

seguimiento de manos desarrollado anteriormente. Estas gafas ofrecen la posibilidad de eliminar cualquier otro sensor o tecnología externa puesto que cuentan con sensores Inside-Out que rastrean los movimientos del usuario sin la necesidad de apoyo externo.

Sin embargo, las Meta Quest cuentan con un inconveniente notable pero manejable, y es que para que el seguimiento de manos funcione correctamente es necesario contar con una buena iluminación en la sala donde se utilicen. Se recomienda una iluminación homogénea y, a poder ser, evitar la luz directa del sol que podría interferir en la captura de los sensores, dando lecturas incorrectas. Las gafas empleadas para el desarrollo del proyecto desde su inicio fueron las Meta Quest 2, sucesoras del modelo Oculus Quest, las cuales también pueden referirse bajo el término ALL-IN-ONE-VR, y predecesoras de los últimos modelos Meta Quest 3 y Meta Quest Pro. Con respecto al nombre del sistema, es importante conocer el contexto. En 2021, cuando Facebook cambió su nombre a Meta, para conseguir una mejor cohesión y simplificar su arquitectura de marca, decidió retirar la marca Oculus para que pasase a llamarse también Meta. Esto genera que el sistema previamente llamado Oculus Quest pasase a llamarse Meta Quest. Como diversos modelos no fueron actualizados ni vendidos posteriormente a este cambio, podemos encontrar sets de gafas antiguos que mantuvieron en su nombre la palabra Oculus.

3.2.1 Oculus Quest

Para uno de los proyectos base de los que se parte, *UnrealTrackingGrasp*¹, las gafas seleccionadas fueron el modelo Meta Quest 2, justificado por ser el set que contaba con todo lo necesario para el desarrollo de su proyecto, integraba la mayoría de funcionalidades que se habían visto hasta la fecha en otros modelos anteriores e incluía los sensores Inside-Out mencionados anteriormente, lo que permitía eliminar la necesidad de utilizar cualquier otro tipo de sensor externo. Las gafas contaban por sí solas con todo lo necesario para operar, sin embargo, se hizo uso de un cable para conectarlas al ordenador ya que agilizaba el proceso de ejecución del proyecto. A pesar de no ser necesarios en nuestro caso, las gafas vienen con dos controladores puesto que la mayoría de juegos o aplicaciones que se pueden ejecutar en las gafas precisan de sus controladores para poder interactuar con las aplicaciones o con el entorno virtual.

Especificaciones Oculus Quest:

- Panel Organic Light Emitting Diode (OLED) 1600x1440px (por ojo)
- Velocidad de refresco: 72Hz
- Conector USB Tipo C
- Tracking: 6 Degrees Of Freedom (DOF)
- Audio integrado en las propias gafas
- Central Processing Unit (CPU): Qualcomm® Snapdragon 835
- Random Access Memory (RAM): 4GB

¹<https://github.com/3dperceptionlab/unrealtrackinggrasp>



Figura 3.1: Gafas y controladores Oculus Quest

3.2.2 Meta Quest 2

El modelo seleccionado para ser usado en este proyecto es Meta Quest 2 Texto principal² (anteriormente Oculus Quest 2). En comparación con su modelo predecesor, las gafas Meta Quest 2 cuentan con mejoras en el visor; sus paneles son Liquid Crystal Display (LCD), lo que genera una mejora en la resolución, nitidez y claridad de imagen. Gracias a todo esto, disminuye la sensación de mareo en los usuarios y pospone la sensación de vista cansada, puntos muy beneficiosos al tener como público objetivo de este proyecto a personas mayores. También se ha rediseñado el visor para hacerlo más atractivo y ligero para poder ser usado por períodos más largos de tiempo.

En cuanto a funcionalidades y especificaciones, es muy similar a las ya mencionadas Oculus Quest; ambas podrían haber sido seleccionadas para el desarrollo del proyecto puesto que los dos sets de gafas cuentan con el sistema de sensores Inside-Out y su capacidad de conectarse al Personal Computer (PC) a través de Quest Link, delegando los costes computacionales al ordenador, que contará con mejores recursos.

Especificaciones Meta Quest 2:

- Panel LCD 1832×1920px (por ojo)
- Velocidad de refresco: 72Hz
- Conector USB Tipo C
- Tracking: 6 DOF
- Audio integrado en las propias gafas
- CPU: Qualcomm® Snapdragon XR2 Platform

²<https://www.meta.com/es/quest/products/quest-2/?srsltid=AfmB0oo8PVnBgJrrh3vhqTaaKrnYnyMBNdREFwWSx1kGKfJAsUvkH7-e>

- RAM: 6GB



Figura 3.2: Gafas y controladores Meta Quest 2

3.2.3 Meta Quest 3

Este set de gafas³ es el último modelo que ha lanzado Meta al mercado. Como su nombre indica, se trata de la evolución de las Meta Quest 2. La mayor novedad, sin duda, es su mejora en la realidad mixta. Se les han introducido mejoras en su sistema de cámaras; ahora cuenta con dos cámaras RGB que recogen información a todo color y con mayor resolución que el modelo anterior. Esta mejora permite obtener una visión real del entorno a la vez que se interactúa con elementos virtuales. Si bien es cierto que las Meta Quest 2 también podían recoger información del entorno, únicamente lo hacían en escala de grises y con una calidad baja. En cuanto a los controladores, se les ha añadido tecnología háptica TruTouch para aportar una sensación más real. También se les ha añadido un seguimiento de los controladores con visión artificial y aprendizaje automático que les permitirá seguir tus gestos.

Especificaciones Meta Quest 3:

- Resolución de pantalla: 2064 × 2208px (por ojo)
- Velocidad de refresco: 72 Hz, 80 Hz, 90 Hz, 120 Hz
- Conector USB Tipo C
- Tracking: 6 DOF
- Audio integrado en las propias gafas
- CPU: Qualcomm® Snapdragon XR2 Gen 2

³https://www.meta.com/es/quest/quest-3/?srsltid=AfmB0oqATHFyICQrRtN9cmxF_cRKDWuVR0fQidCYhT0cafuvzuU22BId

- RAM: 8GB



Figura 3.3: Gafas y controladores Meta Quest 3

Sin embargo, debido al poco tiempo que llevan en el mercado, solo se tenía disponibilidad de ellas una vez que el proyecto ya había comenzado su desarrollo, por lo que no eran una opción viable. Por lo tanto, no era una buena opción escogerlas sin saber tampoco cómo de compatibles serían con el *hand tracking* desarrollado en los proyectos base.

3.2.4 Equipo para el desarrollo

Es necesario trabajar con un ordenador que cumpla unos requisitos mínimos tanto para poder ejecutar UE4 como para el sistema Quest Link, este sistema viene incluido en las gafas de realidad virtual seleccionadas y permite conectarlas al PC y enviar datos entre ellos. Es más, si no se cuenta con al menos la tarjeta gráfica más básica soportada por Meta Quest, no sería posible conectar el dispositivo de realidad virtual al ordenador.

Requisitos de Unreal Engine 4 (UE4)

UE4 es un motor de juego que se ejecuta en tiempo real, lo que significa que para mostrar cada fotograma, el ordenador personal donde se ejecute deberá realizar una serie de cálculos costosos. Por lo tanto, dependerá de nuestro ordenador y sus especificaciones la cantidad de Fotogramas por segundo (FPS) que pueda alcanzar la aplicación. Un número insuficiente de FPS puede provocar problemas como retrasos entre las acciones del usuario y su representación en el juego, saltos en la imagen o incluso mareo y náuseas, afectando negativamente la experiencia del usuario y reduciendo la inmersión en el entorno virtual.

UE4, cuenta con una gran cantidad de funcionalidades y facilidades para los desarrolladores. Sin embargo, estas ventajas van de la mano de requisitos del sistema elevados ⁴. A

⁴https://dev.epicgames.com/documentation/en-us/unreal-engine/hardware-and-software-specifications-for-unreal-engine?application_version=5.4

continuación, se muestran los requerimientos mínimos recomendados para ejecutar UE4.

Requerimientos Unreal Engine	
Operating System	Windows 10 64-bit versión 1909 revisión .1350 o superior, o versiones 2004 y 20H2 revisión .789 o superior
Processor	Quad-core Intel o AMD, 2.5 GHz o más rápido
Video RAM	8 GB RAM
Graphics Card	Tarjeta gráfica compatible con DirectX 11 o 12 con los drivers más recientes

Tabla 3.1: Especificaciones para usar Unreal Engine

Requisitos para usar Meta Quest y Meta Quest 2 en UE4

Para utilizar las gafas Meta Quest y Meta Quest 2 en UE4, es necesario conectarlas a un PC mediante un cable especial. Este cable, debe tener un conector USB tipo C en un extremo (para las gafas) y un USB tipo A o C 3.0 en el otro (para el PC). Además, debe poder alcanzar velocidades de transferencia de mínimo de 5 Gigabits por segundo (Gbps) para garantizar la transmisión en tiempo real entre el PC y las gafas y evitar cualquier tipo de *delay*.

Sin embargo, no solo el cable es importante. Para que la conexión Meta Quest funcione, tanto el sistema operativo como el *hardware* del PC deben cumplir con unos requisitos mínimos establecidos por Meta. Estos requisitos están diseñados para garantizar una experiencia de usuario fluida. El *hardware* debe incluir una Graphics Processing Unit (GPU) potente para procesar gráficos 3D, una CPU capaz de manejar la carga de trabajo asociada a la RV y suficiente memoria RAM para gestionar los procesos necesarios. Asimismo, el PC debe contar con al menos un puerto USB 3.0 compatible.

Cumplir con estos requisitos ⁵ será, ya que de lo contrario no será posible conectar el PC y las gafas de realidad virtual.

Requerimientos Meta Quest Link		
Componente	Especificaciones mínimas	Especificaciones recomendadas
Procesador	Intel i5-4590 / AMD Ryzen 5 1500X o superior	Intel i7 o AMD Ryzen 7
Tarjeta gráfica	NVIDIA GeForce GTX 1060M	Nvidia RTX Serie 20* o AMD Radeon RX Serie 6000
Memoria	8 GB de RAM o más	RAM 16 GB DDR4
Sistema operativo	Windows 10 64-bit	Windows 10 64-bit

Tabla 3.2: Especificaciones para usar Meta Quest Link

⁵<https://www.meta.com/es-es/help/quest/articles/headsets-and-accessories/oculus-link/requirements-quest-link/?srsltid=AfmB0opT0jE0mN05HJiXwNt1X9bQmImxf7EB3M-UpuuvpaaOk-DVMFue>

Especificaciones del PC personal	
Motherboard	Gigabyte Z490 UD
CPU	Intel Core i5 10400F 16GB DDR4 12-Core processor Hasta 4.30 GHz
GPU	NVIDIA GeForce RTX 3060 ti 8GB
RAM Memory	16 GB
Operating System	Windows 10 64-bit

Tabla 3.3: Especificaciones del ordenador personal usado

Ordenador personal usado

Teniendo en cuenta todos los requisitos recién mencionados es fundamental contar con un ordenador con buenas especificaciones, especialmente en los componentes del procesador y la tarjeta gráfica. Para el desarrollo del proyecto, se utilizó un equipo que cumple con estas características, sus especificaciones son las siguientes:

3.3 Software

Como ya se ha observado anteriormente, para el desarrollo de este proyecto se utilizará el motor de juego UE4, que explicaremos en la sección 3.3.1, el *framework* de *hand tracking* creado por Oculus, y que se utiliza en el sistema de agarre desarrollado en los TFG previos del que se parte que se explican en 3.3.2. .

3.3.1 Unreal Engine 4

El motor UE4⁶, creado por Epic Games, destaca por ser una herramienta muy popular en la creación de videojuegos gracias a su versatilidad y potencia, cada vez obtiene mayor reconocimiento en la industria, ya que grandes desarrolladoras lo están empleando. Para nuestro proyecto es ideal puesto que cuenta con todas las funcionalidades que necesitamos, permite integrar un *framework* de seguimiento de manos, y además permite una fácil incorporación de las gafas Oculus. También ofrece la posibilidad de codificar en C++ o usar *Blueprints*, un método visual para programar basado en nodos que facilita y agiliza el proceso de desarrollo con una curva suave de aprendizaje.

En *Blueprints*, los nodos se usan para realizar acciones o funciones específicas, y las interconexiones entre ellos determinan la secuencia y la estructura lógica del proyecto. Podemos encontrar un nodo para cada funcionalidad que podía ofrecer C++. Este tipo de programación basada en nodos y conexiones facilita la organización de la estructura del proyecto y su comprensión. Cada objeto del proyecto se implementa como un *blueprint* basado en clases proporcionadas por el motor, como actores, componentes o animaciones, a los que se pueden

⁶<https://www.unrealengine.com/es-ES>

añadir las funcionalidades deseadas.

Este sistema facilita la creación de los proyectos, permitiendo un desarrollo más ágil y eficiente. Además, cuenta con una interfaz intuitiva que mejora la experiencia del desarrollador durante el diseño y la implementación.

3.3.2 Sistema de agarre y Framework de hand tracking

Como ya se ha mencionado, partimos de un TFG realizado por compañeros de la carrera en años anteriores. Su trabajo dio frutos, consiguiendo desarrollar un sistema de agarre de objetos realista usando el sistema de detección y seguimiento de manos *hand tracking*⁷ que ofrece Oculus. Este *plugin* es soportado por UE en la versión 4.26, versión que se va a utilizar para el desarrollo del escape room.

Comenzar a usar este *plugin* es sencillo. Después de integrarlo en el motor, podremos obtener el esqueleto de las manos moviéndose en tiempo real. Para ello, lo primero será crear un componente **OculusHand** y hacerlo hijo del componente **MotionController**. El sistema permite, además, editar la malla si así se desea, y ofrece la posibilidad de enlazar acciones a gestos predefinidos. Configurando un gesto, se consigue que el jugador se mueva a través del mundo “señaland” con la mano en la dirección que quiere moverse.

Por lo tanto, empleando este *framework*, y citando al autor del TFG previo, se obtuvo “*un sistema robusto que permite al usuario llevar a cabo la interacción objeto-mano dentro del entorno con mucha facilidad*” [21]. El proyecto actual hará uso de esta tecnología, eliminando por completo el uso de controladores y obteniendo una mayor sensación de inmersión. Al emplear este sistema, los objetos dentro del juego actuarán como lo harían en la vida real. Por ejemplo, si un objeto es muy ancho, una única mano no podrá levantarla, pero si se intenta coger por una parte más fina del objeto, al igual que en la vida real, sí será posible.

3.3.3 Blender

Para poder crear y personalizar al cien por ciento el entorno y los puzzles del escape room, será necesario un programa de modelado. Existen una gran cantidad de programas que nos ofrecen las funcionalidades necesarias para lograrlo. Se ha decidido usar Blender⁸ debido a la experiencia previa con él, ya que a lo largo de la carrera se ha utilizado en varias ocasiones y para varias asignaturas. De esta forma, se evita tener que aprender a utilizar un nuevo programa.

Blender es un software de código abierto y gratuito que cuenta con una enorme red de apoyo global. Es uno de los programas más utilizados a la hora de hacer modelados 3D y cuenta con muchas funcionalidades, de las cuales nos interesan el modelado y el texturizado. Además, permite exportar nuestros modelos en varios formatos compatibles con UE4, como .obj para objetos sueltos o .fbx para conjuntos de objetos.

⁷<https://developers.meta.com/horizon/documentation/unreal/unreal-hand-tracking/>

⁸<https://www.blender.org/>

3.3.4 Overleaf

Overleaf es una herramienta en línea que permite escribir y visualizar documentos redactados en LaTeX. El uso de LaTeX garantiza una tipografía clara y una presentación elegante y de gran calidad, ideal para la elaboración de este documento. Overleaf no solo facilita y simplifica todo el proceso de redacción de documentos, sino que también cuenta con funcionalidades como colaboración en tiempo real, plantillas predefinidas y control de versiones.

4 Desarrollo

Este capítulo, en el que se describe la implementación de los sistemas desarrollados y el trabajo realizado, se divide en varios apartados: la sección 4.1, en la que se profundiza en el funcionamiento de UE4; la sección 4.2, donde se detalla el proceso seguido para implementar el sistema de puzzles; la sección 4.3, en la que se describe la implementación de la mecánica principal del *escape room*; las secciones 4.4, 4.5, 4.6, 4.7 y 4.8 en las que se explican distintas elementos y funcionalidades que se han programado para el juego; y finalmente, la sección 4.9, en la que se la implementación de un sistema de seguimiento del usuario.

4.1 Introducción

Anteriormente, en la sección 3.3.1 hemos visto una pequeña introducción a la herramienta visual de programación ofrecida por UE4, conocida como *blueprints*. A la hora de crear un proyecto de RV, UE crea por defecto un proyecto basado en esta herramienta.

UE ofrece un sistema basado en actores, clases y componentes que nos permite crear y definir el comportamiento de los elementos de la escena. Los actores son entidades dentro del mundo del juego que representan cualquier objeto dentro de él. Un actor puede ser un objeto estático o dinámico, y su funcionalidad puede definirse o personalizarse mediante *blueprint*, asignándole comportamientos como moverlo, cambiar su color o detectar colisiones. Un actor puede tener varios componentes. Los componentes que se agregan a un actor dependen de él para existir, es decir, no pueden existir por su cuenta en la escena del juego. Un componente se encarga de agregar funcionalidades extra a un actor. Una **StaticMesh** es un tipo de componente visual que representa un objeto estático en el mundo del juego. Se utiliza para mostrar una malla 3D estática, como un modelo de un objeto. Otro tipo de componente que puede contener un actor son los componentes de colisión, como *BoxCollision* o *SphereCollision*, según la forma del área de colisión que deseemos. Un componente de colisión se encarga de detectar cuando otros actores o componentes entran en su área. Se puede personalizar esta colisión de forma que solo detecte cierto tipo de actores. Cuando esto sucede, el componente cuenta con la capacidad de ejecutar eventos como *OnComponentBeginOverlap* o *OnComponentEndOverlap*. Una ventaja de estos eventos es que devuelven el actor o actores que estén colisionando con el *Box* o *SphereCollision*. Ambos se usarán en gran medida a lo largo de todo el proyecto en todos los sistemas desarrollados, como por ejemplo, al interactuar con objetos, al recoger un objeto o hacer que algo suceda cuando el jugador se acerca a un área específica.

En UE , el flujo de juego se gestiona a través de eventos que se activan en momentos específicos durante el ciclo de vida de un actor o de la propia aplicación. Dentro de cada *blueprint*, podemos asignar las diferentes funcionalidades a eventos que se ejecutarán por sí

solos bajo ciertas condiciones.

Como eventos principales del flujo de juego encontramos:

- **Event BeginPlay:** Este evento se ejecuta una sola vez cuando un actor es creado dentro del entorno, ya sea al inicio del juego o durante su ejecución. Es el momento ideal para inicializar variables o ejecutar acciones que deben ocurrir una única vez, como configurar el estado inicial de un actor o sistema.
- **Event Tick:** Este evento se ejecuta una vez por cada bucle de juego, generalmente a 60 FPS. Se utiliza principalmente para realizar actualizaciones constantes.

4.1.1 Game Instance

La clase **Game Instance** es una entidad que persiste a lo largo de todo el ciclo de vida de la aplicación. Se inicializa al comienzo de la ejecución del juego y no se destruye al cambiar o pasar de nivel. Esto la convierte en un contenedor global ideal para almacenar datos y lógica que deben mantenerse constantes durante toda la sesión de juego.

Además, una de sus principales ventajas es que permite el acceso a su contenido desde cualquier punto del proyecto, proporcionando una solución eficiente para la gestión de datos globales. Para aprovechar estas características, se ha diseñado una clase **GameInstance** personalizada a la que se ha llamado **MainGameInstance**. Una cualidad de esta clase es que es accesible desde cualquier parte del juego y en cualquier momento de la ejecución de la aplicación.

Cuando se construye, durante el evento **Init**, se instancian las clases **PuzzleManager** y **GameDebugger**, ambas derivadas de la clase base *UObject*. Estas clases no son actores, por lo que no tienen una representación física en la escena; en su lugar, se comportan como objetos puramente lógicos destinados a gestionar funcionalidades específicas del juego.

4.2 Sistema de Puzzles

4.2.1 PuzzleManager

PuzzleManager es la clase encargada de gestionar todos los puzzles del *escape room*. Al ser una lógica de carácter global, no específica de ningún nivel, el **PuzzleManager** se encuentra integrado en la clase **MainGameInstance**, lo que permite que sus funcionalidades sean accesibles desde cualquier parte del proyecto. De esta forma, para acceder al **PuzzleManager** solo habrá que llamar primero a **MainGameInstance**.

PuzzleManager almacena y gestiona dos variables implementadas como mapas clave-valor: **PuzzleStorage** y **PendingPuzzleObjects**.

- **PuzzleStorage:** Este mapa almacena los puzzles activos en el nivel donde se encuentra el jugador. Cada entrada utiliza una clave única de tipo *string* que sirve como identificador

del puzzle, asociada a una estructura personalizada (*Struct*). En UE, los valores de un mapa pueden ser de tipos básicos (enteros, booleanos, cadenas) o más avanzados (vectores, referencias a objetos, enumeraciones o estructuras personalizadas). En este caso, se ha diseñado una estructura personalizada que permite almacenar un *array* de los objetos `PuzzleObjects` que contiene el puzzle.

- `PendingPuzzleObjects`: Este mapa utiliza una estructura similar, pero se destina a almacenar aquellos `PuzzleObjects` que se hayan creado sin que el puzzle del que dependen esté definido o cargado. De esta forma, se garantiza la integridad de los datos, permitiendo que los objetos pendientes sean procesados adecuadamente una vez que su puzzle correspondiente se haya registrado en el sistema.

4.2.2 BP_PuzzleObject

La clase `BP_PuzzleObject` hereda de *Scene Component*. Esto implica que se trata de un componente que añade lógica o funcionalidad al actor al que esté asociado. Sin embargo, no puede existir de manera independiente dentro del nivel.

Cuenta con dos variables:

Nombre de la variable	Tipo de dato	Descripción
Tag del puzzle	String	Indica el puzzle al que pertenece y del que depende.
Estado de resolución	Booleano	Indica si el objeto está solucionado o no.

Tabla 4.1: Variables en `BP_PuzzleObject`

Cuando se crea un `BP_PuzzleObject`, se accede al `MainGameInstance` para obtener así acceso a el `PuzzleManager`. Dentro del `PuzzleManager` recorre el mapa `PuzzleStorage` en busca de un puzzle con el mismo `tag` que el objeto recién creado. Si lo encuentra, añade el objeto al `array` correspondiente. En caso contrario, lo agrega al mapa `PendingPuzzleObjects` junto con su `tag`.

4.2.3 BP_Puzzle

Al igual que `BP_PuzzleObject`, `BP_Puzzle` es un *Scene Component* que se añadirá a un actor para añadirle lógica.

Cada puzzle está formado por:

Nombre de la variable	Tipo de dato	Descripción
BP_PuzzleObjects	Array	Contiene todos los objetos del tipo <code>BP_PuzzleObject</code> que componen el puzzle.
Tag	String	Identificador único del puzzle.
Puzle Resuelto	Booleano	Indica si el puzzle completo está solucionado.

Tabla 4.2: Variables en `BP_Puzzle`

Un puzzle se considerará solucionado cuando todos los `PuzzleObjects` que contiene estén marcados como tal.

Cuando se crea un `BP_Puzzle`, sigue un proceso similar a la creación de un `BP_PuzzleObject`. Accede al `PuzzleManager` para poder añadir este nuevo puzzle al mapa `PuzzleStorage`. Dentro de `PuzzleManager`, antes de registrar un puzzle, se verifica que no exista ya uno con el mismo identificador. Además, comprobará si hay objetos dentro del mapa `PendingPuzzleObjects` con ese mismo `tag`, en caso positivo los agregará al *array* de `BP_PuzzleObject` del puzzle correspondiente dentro del mapa `PuzzleStorage`. Añadir un puzzle no significa únicamente registrar su `tag`, sino también el *array* que contiene todos los `BP_PuzzleObjects` que contiene.

4.2.4 Flujo de ejecución de puzzles

Durante la ejecución del nivel cada `BP_Puzzle`, realiza verificaciones periódicas, cada 0,2 segundos, para determinar si está resuelto. Recorre su *array* de `PuzzleObjects` para determinar si todos están completados.

Si se detecta que el estado del puzzle ha cambiado (de no resuelto a resuelto o viceversa), se ejecutan los métodos correspondientes:

- `OnPuzzleCompleted`: Se activa cuando un puzzle se completa por primera vez.
- `OnPuzzleUncompleted`: Se ejecuta cuando un puzzle, previamente resuelto, deja de estarlo.

En estos métodos se implementa la lógica específica que debe ocurrir en cada caso, como la activación de animaciones, reproducción de efectos sonoros o entregarle una recompensa al jugador. En el juego sucederán estas tres cosas en distintos momentos. Cuando se resuelva cualquier puzzle se mandará la orden de reproducir un sonido para otorgarle al jugador federación de que lo ha completado correctamente. En caso de ser el puzzle final de un nivel, además del sonido, también se mandará la orden de ejecutar una secuencia que animará la puerta de la sala para abrirse indicando que ha completado todo el nivel y que puede salir. Para el resto de puzzles de dentro de cada nivel se le entregará al jugador como recompensa una llave que podrá usar más adelante, en el capítulo de Diseño del Escape Room se explica en detalle para qué servirá y el funcionamiento de cada puzzle.

Es importante cada vez que se carga un nivel llamar al método `Clear` del `PuzzleManager` para asegurar una correcta gestión de los datos. De esta forma se consigue:

- Evitar datos obsoletos de puzzles cargados de niveles anteriores que no son relevantes en el nivel actual. Pudiendo llegar a generar puzzles duplicados o confundir la lógica de eventos.
- Evitar sobrecarga de memoria por mantener datos de puzzles de niveles anteriores.
- Evitar referencias invalidas al tratar de acceder a puzzles u objetos que se mantienen registrados pero que no se encuentran en el nivel actual.

4.3 Sistema de interacción

La mecánica principal que se usará para resolver los puzzles en el *escape room* será la colocación de objetos en un lugar específico. Para ello se ha desarrollado un sistema de interacción. Este sistema está formado por tres elementos claves, en este apartado se describen especificando su funcionalidad y las relaciones entre ellos.

1. **BP_InteractiveObject:** Representa los objetos con los que el usuario podrá interactuar, que pueden ser recogidos, movidos o colocados.
2. **BP_ObjectPlacement:** Define aquel lugar dónde un BP_InteractiveObject puede ser colocado.
3. **BP_PlacementPoint:** Es un componente asociado al BP_ObjectPlacement que determina qué BP_InteractiveObject puede ser colocado en él, además de el cómo y dónde específicamente en el espacio.

A continuación, se detalla cada uno de estos elementos.

4.3.1 BP_InteractiveObject

BP_PickupObject es una clase creada en el proyecto previo del que partimos que contiene toda la lógica que especifica que un objeto puede ser recogido y transportado con las manos. **BP_InteractiveObject** hereda de la clase **BP_PickupObject**, de esta forma contiene toda su lógica, pero además se le pueden añadir funcionalidades extra. Esta lógica adicional incluye:

- La capacidad de ser colocado.
- Un identificador de tipo denominado *Interactive ID*.

Cada instancia de **BP_InteractiveObject** contiene:

- Una malla (*mesh*) que define su aspecto visual y colisión. En ocasiones, interesaría crear una colisión personalizada para facilitar la interacción o reducir el coste computacional en los cálculos de colisión. UE4 si no detecta que exista previamente una colisión genera una automáticamente.
- Tres variables propias:

Nombre	Tipo	Descripción
Interactive ID	Enumerado	Identificador que indica el tipo de objeto interactivo.
IsPlaced	Booleano	Indica si el objeto se encuentra colocado.
PlaceObject	BP_ObjectPlacement	Referencia al ObjectPlacement donde está colocado el objeto interactivo.

Tabla 4.3: Variables en BP_InteractiveObject.

Además, ciertos objetos pueden incluir un identificador genérico opcional útil a la hora de resolver puzzles.

4.3.2 BP_PlacementPoint

El **BP_PlacementPoint** es un componente que se añade a un **BP_ObjectPlacement**. Se representa visualmente como un punto asociado al **ObjectPlacement**. Según su posición relativa al **BP_ObjectPlacement** determinará dónde se colocará el **InteractiveObject**. Este componente define a través de una variable enumerada (**InteractiveID**) qué tipo de **InteractiveObject** puede ocupar su lugar.

Nombre	Tipo	Descripción
Interactive ID	Enumerado	Define el tipo de objeto interactivo que puede ocupar este punto.
HasObjectPlaced	Booleano	Indica si hay un objeto interactivo colocado ocupando el lugar del punto.
ObjectPlaced	BP_InteractiveObject	Si existe, referencia al BP_InteractiveObject actualmente colocado en este punto.

Tabla 4.4: Variables en BP_PlacementPoint.

4.3.3 BP_ObjectPlacement

El **BP_ObjectPlacement** define las áreas donde los objetos interactivos pueden ser colocados. Puede incluir o no una malla visual, pero debe contener obligatoriamente un **InteractiveBoxCollider**, cuya función es detectar si un **InteractiveObject** está colisionando con el área de colocación.

- Cada instancia de **BP_ObjectPlacement** gestiona un mapa y dos arrays.
 - *Array LastOverlapedActors*: Contiene los actores que se encontraban colisionando en el último *frame*.
 - *Array OverlapsToErase*: Contiene los objetos que han dejado de colisionar.

Un **BP_ObjectPlacement** puede tener varios **BP_PlacementPoints**, permitiendo guardar múltiples objetos en su interior. De cada **PlacementPoint** se registra:

- Su posición relativa al **ObjectPlacement**.
- Su vector de transformación, que determina dónde y cómo se colocará el objeto.

El BP_ObjectPlacement gestiona un mapa clave-valor que organiza todos sus PlacementPoints por tipo de objeto interactivo (Interactive ID), donde:

- **Clave:** Interactive ID.
- **Valor:** Array de PlacementPoints con ese identificador.

Al iniciar el juego (Event BeginPlay), el BP_ObjectPlacement recorre todos sus hijos. Para cada hijo que sea un PlacementPoint, lo almacena en el mapa bajo el Interactive ID correspondiente. Si no existe la clave, se crea una nueva entrada.

4.3.4 Funcionamiento en tiempo de ejecución/Flujo de ejecución

- **Colocación de objetos interactivos** Durante el juego, el ObjectPlacement realiza las siguientes comprobaciones:
 1. **Detección de colisión:** Utiliza el InteractiveBoxCollider para detectar todos los InteractiveObjects colisionando. Comprueba con respecto al *frame* anterior si hay nuevos InteractiveObjects o, si por el contrario, alguno ha cesado el contacto. Aquellos objetos que se encontrasen colisionando pero actualmente no lo estén, se agregan al array OverlapsToErase para eliminarse. El array LastOverlapedActors, se rellena con todos los objetos interactivos que se encuentren colisionando, sin importar su estado anterior, puesto que en el próximo *frame* todos estos objetos se usarán para la comprobación del cambio de estado de colisión.
 2. **Condiciones para colocar un objeto:** El objeto debe tener la variable Is Picking en *false*, indicando que no está siendo sostenido por el jugador. Además, el ObjectPlacement debe encontrar un PlacementPoint válido (con el mismo Interactive ID y HasObjectPlaced en *false*).
 3. **Proceso de colocación:**
 - El ObjectPlacement recorre todos sus PlacementPoints hasta dar con el primero que cumpla las condiciones.
 - Se asignan las transformaciones del PlacementPoint al objeto, actualizando su posición y rotación global.
 - Se actualizan las variables:
 - * **IsPlaced:** *true* en el objeto interactivo.
 - * **HasObjectPlaced:** *true* en el PlacementPoint.
 - * Se almacena una referencia al objeto en el PlacementPoint.
 - Se llama al método OnObjectPlaced del objeto interactivo, donde se pueden definir reacciones como reproducir sonidos, cambiar colores, activar eventos, etc.
 - **Retiro de objetos interactivos** Cuando un objeto interactivo es retirado (por el jugador o al dejar de colisionar con el ObjectPlacement):
 - Se actualizan las variables:

- * `HasObjectPlaced`: *false* en el `PlacementPoint`.
- * `ObjectReference`: Se elimina en el `PlacementPoint`.
- * `IsPlaced`: *false* en el objeto interactivo.
- * `PlaceObject`: Se elimina la referencia al `ObjectPlacement` en el `InteractiveObject`.
- Se llama al método `OnObjectUnplaced`, que permite realizar acciones personalizadas, como invertir los efectos realizados al colocarlo.

4.4 Candado numérico

Otra mecánica que se ha programado para los puzzles del escape room es un candado que se resuelve con una combinación numérica específica. Los puzzles basados en códigos numéricos son elementos clásicos en los *escape rooms* y es muy común encontrarse con un puzzle donde, en el nivel, haya un número oculto que sea la solución al puzzle.

Se ha creado un objeto con forma de rueda de 10 lados; a cada lado se le ha asignado un *widget* con un número del 0 al 9. El jugador podrá interactuar con él haciendo rodar la rueda hacia arriba y abajo para pasar de un número a otro. Si la `BoxCollision` de la rueda detecta al jugador colisionando con ella, almacena la posición de la mano y, una vez esta suba o baje lo suficiente, irá variando su rotación según la dirección del movimiento. Una variable editable desde el *viewport* determinará qué número deberá estar en la zona frontal para considerarse solucionado. Cada vez que el usuario haga rodar la rueda, se accederá a la rotación global de esta para determinar si la cantidad de rotación que presenta coincide con la indicada por la solución.

Para formar un candado completo, únicamente hará falta colocar varias de estas ruedas juntas. Cada rueda será un elemento del puzzle, y, por lo tanto, únicamente cuando todas se consideren solucionadas, el candado al completo se considerará resuelto.



Figura 4.1: Ejemplo de candado numérico dentro del *escape room*

4.5 Material Highlight

Un material es el encargado de indicarle al motor de renderizado exactamente cómo se va a ver la superficie de un objeto y cómo deberá interactuar con la luz dentro de la escena. Se encarga de definir todos los aspectos de la superficie, lo que incluye el color, la reflectividad, rugosidad, transparencia, etc.

La instanciación de materiales[1] en UE4 se usa para cambiar la apariencia de un material sin incurrir en una recopilación costosa del material. En UE4 hay disponibles dos tipos de instancias de material: constante y dinámica.

- **Constante:** se calcula únicamente antes del tiempo de ejecución.
- **Dinámica:** se puede calcular y editar en tiempo de ejecución.

Por lo tanto, un material dinámico permite, en mitad del juego, cambiar completamente el aspecto de un objeto gracias a la posibilidad de modificar parámetros del material. En nuestro caso, hemos creado un material dinámico, `M_Highlight`, que se le aplica a los `BP_InteractiveObjects`. Este material crea sobre el contorno de un objeto cercano un brillo amarillo para indicarle al jugador con qué objetos puede interactuar. De otra forma, podría llegar a ser muy confuso y hacerle perder al usuario mucho tiempo tratando de recoger todos los objetos del juego. Este material otorga entonces al jugador una guía hacia los puzzles.

La lógica detrás de esto es que, cuando el `HandController` detecte en un área cercana un `BP_InteractiveObject` (usando un `SphereBoxCollision` añadido como componente a las manos), accederá al material del objeto y cambiará un valor de este, activando el efecto `Highlight`.

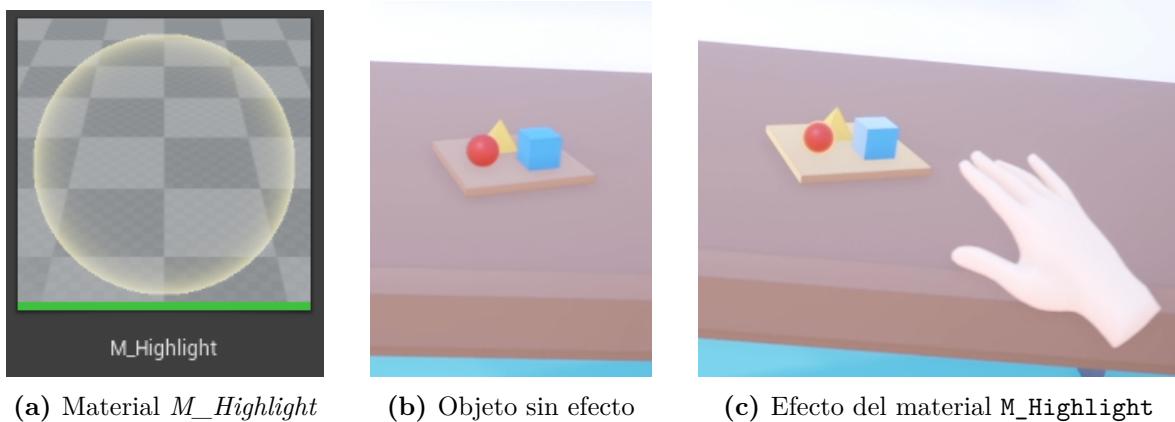


Figura 4.2: Resultado del efecto `Highlight` para los objetos interactivos

4.6 Pistas

En situaciones donde el jugador se vea atascado y no logre resolver un acertijo, o cuando por cualquier motivo necesite algún tipo de ayuda para resolver un puzzle, se ha desarrollado

una funcionalidad como medida preventiva frente a la frustración y el abandono. Se ha añadido la capacidad de activar pistas dentro del entorno del juego. Al hacerlo, aparecerá una imagen que tendrá una pista escrita sobre cómo resolver un puzzle específico. No solo ofrece esta ayuda, sino que, aparte, la posición donde aparece la pista ubica dónde está centrado el puzzle. Para asegurarse de que el usuario llegue a observar las pistas cuando las necesite, estas se implementan utilizando ***Billboards***.

Un *Billboard* es un elemento que siempre estará orientado en dirección a la cámara, es decir, hacia el usuario. Conforme el jugador se mueve, el *Billboard* se orientará automáticamente para estar siempre enfocado hacia él, dando una sensación de estar “siguiéndolo con la mirada”. En UE4, los *Billboards* se gestionan mediante objetos 2D que utilizan una textura para determinar su aspecto. Gracias a no ser elementos 3D, los *Billboards* son muy eficientes en cuanto a rendimiento, son poco exigentes en cuanto a recursos, ya que no requieren cálculos complejos.

Al objeto *Billboard* se le añade un componente de *SphereCollision* con una colisión personalizada, de manera que solo detecte al jugador. Cuando esto pase, queremos que lance un evento de activar su visibilidad para garantizar que el jugador esté cerca.



Figura 4.3: Ejemplo de una pista sobre el puzzle de poner la mesa 5.3

4.7 Sonido

UE4 permite integrar audio en el proyecto de una forma muy sencilla. Cuenta con un nodo **Play Sound on Location** que reproduce un sonido seleccionado en un lugar específico del mundo 3D. Este nodo permite la configuración de parámetros como editar el volumen, la escala de tono del sonido o el segundo del archivo desde el que se quiere iniciar la reproducción del audio. En el proyecto, el nodo se usará para reproducir sonido en la posición donde se encuentra el jugador, reforzando la sensación de inmersión de la experiencia virtual.

Además, se emplea el mismo nodo para gestionar la música ambiente del nivel, el audio se configura en modo *loop* para que se reproduzca de manera continua y en bucle. Se uti-

lizará una música suave que no distraiga al jugador, pero que sí cree una atmósfera agradable.

Por otra parte, el sistema de audio se utiliza para proporcionar *feedback* positivo al usuario durante la resolución de los acertijos. Cada vez que se complete un puzzle, se reproducirá un sonido que exprese victoria, y cuando se complete el nivel por completo, se reproducirá un sonido de mayor intensidad para reforzar el progreso al jugador.

Finalmente, se han implementado efectos de sonido específicos para eventos clave, como la teletransportación entre niveles. Este sonido es acompañado visualmente por un fundido en negro para una transición entre niveles menos abrupta.

4.8 Reloj

Un reloj virtual es una herramienta de gamificación muy útil para contabilizar tiempos en actividades. Además, ayuda a evitar la desorientación temporal del usuario, un problema común al entrar a un entorno de RV por el efecto inmersivo. Se decidió colocar este reloj virtual en la muñeca, a modo de imitación de un reloj convencional, de modo que la hora pueda consultarse de manera rápida y natural.

Lo primero que hay que realizar para que se visualice el reloj es crear un *UserWidget*. Este es un componente de UE4 que se utiliza para crear elementos de interfaz de usuario; permite diseñar y programar elementos visuales e interactivos, además de colocarlos en pantalla.

Para que el reloj mantuviese su posición en la muñeca incluso cuando las manos se movieran, al crear las manos en el `VRCharacter` se creó una clase hija denominada `BP_LeftHandUI`. La cual contiene como componente el *User Widget* previamente mencionado. Al ser un actor hijo, `BP_LeftHandUI` hereda las transformaciones de su clase padre, es decir las manos, de esta forma, su posición será relativa a ellas.

Poder ver la hora en el juego se logra, por una parte, haciendo uso del `Event Tick`. Este evento se ejecuta aproximadamente cada segundo y llamando a la función `Now` del motor podremos obtener, en tiempo real, la hora y los minutos actuales del sistema. Estos valores se asignan a una variable llamada `CurrentTime`. El diseñador de los *widgets* permite añadir elementos de tipo *Text*. Este elemento *Text* se enlazará a la variable `CurrentTime`, gracias a este enlace la interfaz se actualiza automáticamente mostrando cada vez que la hora, y a su vez, la variable cambien. De esta forma se elimina la necesidad de tener que llamar constantemente a un método para recibir el valor de la hora y actualizar la interfaz.



Figura 4.4: Reloj en tiempo real dentro del juego

4.9 Sistema de Seguimiento

Sabemos que el deterioro cognitivo afecta la velocidad de procesamiento de la información, lo que conlleva a un ritmo más lento en la toma de decisiones. En este contexto, nos interesa poder monitorizar el avance en el *escape room* de un jugador. Con este objetivo en mente, se ha desarrollado un sistema de seguimiento.

Este sistema permitirá monitorizar el desempeño individual de los participantes y se encarga de almacenar datos fundamentales para que, posteriormente, un profesional pueda analizar el estado cognitivo del usuario y su evolución. A través del registro de métricas como el tiempo requerido para resolver los puzzles y la posición y orientación del usuario durante el juego, se obtiene una visión detallada de su comportamiento y progreso. Al final de la ejecución del juego, se generará un fichero JSON que contendrá todos los datos observados.

4.9.1 Integración de Blueprints y C++

A pesar de que la lógica principal del proyecto se desarrolla en *blueprints*, UE4 permite combinarlo junto con *C++*. De esta forma, se ha utilizado *C++* para crear, leer y escribir los ficheros JSON. Para organizar los datos y conseguir un archivo JSON bien estructurado, se ha hecho uso de un *plugin* externo llamado **JSON Blueprints**, disponible en la tienda de UE, que se puede instalar en el motor.

4.9.2 Implementación en C++

El desarrollo sigue la estructura habitual en *C++*, con un archivo *.h* donde se declaran los métodos y uno de extensión *.cpp* donde se implementa la lógica.

En estos archivos, se crea la clase **UFileManager**, que hereda de **UBlueprintFunction Library**. Esto indica que está diseñada para proporcionar funciones que pueden ser ejecutadas en un *blueprint* o en el *Level Blueprint Graph*. Estas funciones son reconocibles porque están etiquetadas con **UFUNCTION(BlueprintCallable)**, lo que las hace accesibles directamente desde los *blueprints*.

La clase **UFileManager** cuenta con tres funciones principales para las cuales se incluye la librería **json.h**, que viene integrada con UE4 y proporciona herramientas para manejar datos en formato JSON:

- **CreateLogFile:** Se encarga de crear un fichero con el nombre y en la ruta especificados por los *FString* que recibe como parámetros. Para ello, utiliza la función `SaveStringToFile`, enviando un *string* vacío para crear el archivo sin contenido inicial.

Parámetro	Descripción
<code>FilePath</code>	<i>FString</i> - Ruta del directorio donde se creará el archivo.
<code>FileName</code>	<i>FString</i> - Nombre del archivo que se desea crear.

Tabla 4.5: Parámetros de `CreateLogFile`

- **WriteToFile:** Se encarga de escribir en un fichero específico utilizando la función `SaveStringToFile` con los datos que recibe como parámetros. También recibe la ruta del archivo y permite controlar si los datos deben sobrescribirse o añadirse al fichero.

Parámetro	Descripción
<code>FilePath</code>	<i>FString</i> - Ruta del directorio donde se encuentra o se creará el archivo.
<code>FileName</code>	<i>FString</i> - Nombre del archivo en el que se escribirá.
<code>Data</code>	<i>FString</i> - Contenido que se desea escribir en el archivo.
<code>Overwrite</code>	booleano - Indica si se debe sobrescribir (<code>true</code>) o agregar (<code>false</code>).

Tabla 4.6: Parámetros de `WriteToFile`

- **ReadFromFile:** Esta función accede a un fichero, lo lee y devuelve todo su contenido como una cadena utilizando la función `LoadFileToString`. La ruta del fichero a leer se recibe como parámetro de entrada.

Parámetro	Descripción
<code>FilePath</code>	<i>FString</i> - Ruta del directorio donde se encuentra el archivo.
<code>FileName</code>	<i>FString</i> - Nombre del archivo que se desea leer.

Tabla 4.7: Parámetros de `ReadFromFile`

4.9.3 Estructuras de almacenamiento de datos

En el proyecto contamos con distintas estructuras personalizadas, diseñadas para organizar y almacenar los datos que posteriormente se registrarán en un archivo JSON. Cada estructura corresponde a un elemento específico del JSON.

- **Object Interactions Structure** Esta estructura almacena información relevante sobre las interacciones del usuario con los objetos interactivos. Los campos que incluye son:

Nombre	Tipo	Descripción
ObjectName	Cadena	Nombre del objeto interactivo.
Hand	Cadena	Indica si el objeto fue sujetado con la mano izquierda o derecha.
TimeOnPickup	DateTime	Momento exacto en que el usuario comienza a sujetar el objeto.
TimeOnDrop	DateTime	Momento exacto en que el usuario suelta el objeto.
Duration	Float	Diferencia entre TimeOnDrop y TimeOnPickup, que representa el tiempo total que el objeto estuvo sujetado.

Tabla 4.8: Campos de Object Interactions Structure

- **Object Interaction Array Structure** Es una estructura que contiene un *array* de elementos del tipo Object Interactions Structure, permitiendo almacenar múltiples interacciones relacionadas con diferentes objetos.

- **Headset Structure** Esta estructura guarda información sobre la posición y orientación del jugador en un momento determinado. Sus campos son:

Nombre	Tipo	Descripción
TimeStamp	Cadena	Marca temporal que indica el momento en que se toman las mediciones.
Position	Vector	Ubicación del jugador en el mundo en el instante especificado por TimeStamp.
Orientation	Vector	Dirección hacia la que está mirando el jugador en ese momento.

Tabla 4.9: Campos de Headset Structure

- **Level Structure** Almacena información general sobre un nivel y sus elementos. Incluye los siguientes campos:

Nombre	Tipo	Descripción
LevelName	Cadena	Nombre del nivel.
HoraInicio	Cadena	Hora en que el jugador entra al nivel.
Time	Float	Duración total de tiempo que el jugador permanece en el nivel.
Puzzles	Array de <code>Puzzle Structure</code>	Información sobre los puzzles y sus datos específicos dentro del nivel.
Headsets	Array de <code>Headset Structure</code>	Registra las mediciones del casco a lo largo del nivel.
ObjectInteractionsMap	Mapa (clave-valor)	Relaciona nombres de objetos interactivos (clave) con <i>arrays</i> de <code>Object Interactions Structure</code> (valor).

Tabla 4.10: Campos de Level Structure

4.9.4 Gestión de datos y creación del archivo de registro

Al inicio de la ejecución del juego, tras instanciar la clase `GameDebugger`, se procede a generar el nombre y la ruta del archivo donde se almacenarán los datos recolectados durante la sesión.

- **Generación del nombre del archivo**

El nombre se genera utilizando la hora actual en el formato *hora_minutos_segundos*. Para evitar posibles errores al manipular el archivo en diferentes sistemas operativos, se sustituyen los caracteres especiales y los espacios por el carácter de subrayado (`_`).

- **Determinación de la ruta del archivo**

La ruta se obtiene concatenando los siguientes elementos:

1. El directorio base del proyecto, obtenido mediante el nodo `Unreal Project Dir`.
2. Una subcarpeta llamada `Logs`.
3. La fecha actual en el formato *day_month_year*.

La combinación de estos elementos asegura que la ruta generada sea compatible con el sistema operativo, utilizando los separadores correctos (/ o \).

- **Creación del archivo**

Una vez generados el nombre y la ruta, se llama al método `CreateLogFile` (implementado en C++) para crear el archivo. Al mismo tiempo, se inicializa una variable denominada `InitApplicationTime`, que almacena la hora y la fecha en el momento de ejecución. Esta variable será utilizada más adelante, cuando se cierre la sesión, para calcular el tiempo total de ejecución de la aplicación.

4.9.5 Gestión de niveles y almacenamiento de datos

Tras crear el archivo de registro en formato JSON, se inicializa el primer nivel del juego y se invoca un método encargado de gestionar la estructura de datos asociada al nivel. Este método no solo se ejecuta al inicio del juego, sino también cada vez que se carga un nuevo nivel.

- **Gestión de datos de niveles anteriores**

1. Si el jugador proviene de otro nivel, se accede al nivel anterior para recuperar todos los datos recolectados durante su ejecución.
2. Estos datos se convierten en un objeto JSON mediante el método `CreateJSON`, que organiza toda la información en un formato adecuado.
3. El objeto JSON generado se convierte a un formato legible con tabulaciones (*pretty string*) y se escribe en el archivo de registro utilizando los métodos `CreateLogFile` y `WriteToFile`.

- **Inicialización del nuevo nivel**

En el caso de que sea el primer nivel del juego, se omite el paso anterior y se procede directamente a generar una nueva instancia de la estructura `Level Structure` para el nivel actual, donde se almacenarán todos los datos recolectados durante la ejecución del nivel.

Esta estructura se inicializa con los datos disponibles, que en este punto son únicamente el nombre del nivel y la hora de inicio, establecida con la hora actual.

Finalmente, la estructura del nivel se añade al conjunto de niveles del juego para su posterior actualización durante la ejecución.

4.9.6 CreateJSON

El método `Create JSON` es responsable de generar toda la estructura de datos en formato JSON. Su objetivo principal es obtener un objeto JSON que contenga todos los datos registrados de los niveles jugados. Estos datos, almacenados en cada estructura `Level Structure`, incluyen:

- El nombre de cada nivel.
- La duración del juego en cada nivel.
- Los tiempos de resolución de todos los puzzles.
- Los datos relacionados con los objetos con los que se interactuó a lo largo del nivel.
- La trayectoria del jugador (posición y orientación, registrada cada 3 segundos).

4.9.7 Estructura JSON

JavaScript Object Notation (JSON) es un formato estructurado, ligero y ampliamente utilizado para almacenar y transmitir datos. Su estructura se basa en pares clave-valor, donde la clave es siempre un *string* mientras que el valor puede ser de diferentes tipos: *string*, números enteros, números decimales, arreglos o incluso otros objetos JSON. Un objeto JSON se define mediante llaves y contiene una colección de pares clave-valor. Su simplicidad y legibilidad hacen que sea fácil de interpretar tanto por humanos como por máquinas.

El objetivo final de este método es obtener un archivo JSON que se estructure como se puede observar en la figura 4.5

4.9.7.1 Construcción de la estructura JSON

Para construir el archivo JSON, se utiliza aquí el *plugin* externo JSON Blueprints que facilita la creación y manipulación de los elementos necesarios. Este *plugin* proporciona nodos específicos que permiten estructurar los datos de manera ordenada. De forma que obtendremos un `JsonObject` por cada objeto, es decir, un `JsonObject` por cada nivel, y dentro de este, otros `JsonObject` por cada puzzle resuelto, por cada objeto con el que se ha interactuado y por cada medida del `headset` guardada.

Un `JsonObject` estará formado por distintos campos con un valor cada uno. Este valor puede llegar a ser otro `JsonObject` (como pasa con el campo de objetos con los que se interactúa, cuyo valor será un `JsonObject` compuesto por un *array* de todas sus interacciones).

- **Creación de valores JSON:** Se generan valores JSON (`JsonValue`) específicos para cada tipo de dato, utilizando los nodos correspondientes del *plugin*.
 - `JSON MAKE String`
 - `JSON MAKE Float`
 - `JSON MAKE Array`
 - `JSON MAKE Object`

Estos valores se rellenan con las métricas almacenadas en el `Level Structure` del nivel actual.

- **Asignación de valores a campos:** Los valores generados se asignan a claves específicas mediante el nodo `JSON Make Field`. Este nodo permite:
 - Definir el nombre del campo (clave).
 - Asignar el `JsonValue` al `JsonObject` correspondiente.
 - Estrategia de construcción jerárquica

La construcción del `JsonObject` sigue un enfoque jerárquico, comenzando por los elementos más pequeños o hijos. Este orden asegura que todos los datos necesarios estén listos antes de integrarse en niveles superiores.

Se recorren todos los niveles registrados. De cada nivel, se extraen los datos almacenados en el `Level Structure`. Cada elemento se convierte en un `JsonValue` del tipo adecuado y

se integra como un `JSONField` dentro del `JsonObject` del nivel.

Se crea un `JsonObject` que contendrá todos los puzzles resueltos, para los cuales se crea un `JSONField` para cada dato del puzzle. Se aplica un proceso similar para las marcas de tiempo del `headset`, con un `JsonObject` padre que agrupa todas las medidas guardadas en `JSONFields`. Sin embargo, para las interacciones con objetos, se seguirá un proceso algo diferente, creando un `JsonObject` que contendrá todos los objetos. Para cada objeto, se creará otro `JsonObject` individual, y se recorrerá su *array* de interacciones. Los datos de cada interacción se almacenarán en distintos `JSONField`.

4.9.8 Métodos de toma de medidas

En este apartado se describen las funciones que contiene `GameDebugger`, utilizadas para almacenar los datos relevantes del sistema. Cada método se activa en momentos específicos durante la interacción del usuario con el entorno virtual.

- **Puzzle Start:** Este método se ejecuta en el momento en que el usuario colisiona con `PuzzleStartTimer`. `PuzzleStartTimer` se crea heredando de `TriggerBox`. Los `TriggerBox` son volúmenes invisibles en el juego que se utilizan para detectar la entrada, permanencia o salida de los actores. Esta función tiene como objetivo almacenar el tiempo inicial del puzzle que acaba de comenzar. Una vez que se detecta la colisión, el sistema registra la hora actual como el inicio del conteo de tiempo para ese puzzle.
- **Puzzle Completed:** Esta función se llama cuando el usuario completa un puzzle. Al activarse, calcula la duración del tiempo que el usuario ha tardado en resolverlo. Para ello, resta al tiempo actual la marca temporal almacenada como tiempo inicial del puzzle. Posteriormente, los datos del puzzle resuelto se agregan al *array* de puzzles del nivel actual.

Tras esta actualización, se invoca el método `CreateJSON` para obtener el `JSON Object` actualizado y el método `SaveDataToFile`, donde este `JSON Object` se convierte a un formato legible (*pretty string*). Este *pretty string* se escribe en el archivo utilizando el método `WriteToFile`, consiguiendo así una versión actualizada del nivel en formato JSON.

De esta manera, el sistema no solo escribe los datos en el archivo JSON al finalizar un nivel completo, sino también al resolver cada puzzle. Este enfoque asegura que, incluso en caso de errores o cierres inesperados de la aplicación, se guarde la mayor cantidad posible de información.

- **Update Headset Data:** El encargado de llamar a este método es el `VRCharacter`, que será el responsable de implementar la lógica necesaria para capturar de forma periódica la posición y orientación de la cabeza del usuario, representada en el juego por la cámara. Por lo tanto, cada tres segundos, envía al método `UpdateHeadsetData` los datos recogidos. Este método se encarga de crear una estructura de tipo `HeadsetStruct`, que almacena la información de posición y orientación cada vez que es llamado, y la agrega al *array* de `Headset Structures` del nivel actual.

- **Add Object Interact:** Esta función es llamada desde el método `PickUpObject` o `ReleaseActor` del `HandController`, dependiendo de si un objeto ha sido recogido o soltado. En ambos casos, el método recibe los siguientes parámetros:
 - El nombre del objeto.
 - Una enumeración que identifica qué mano (izquierda o derecha) interactuó con el objeto.
 - Una variable booleana `isPickUp` que indica si el objeto ha sido recogido (`true`) o soltado (`false`).

Una vez dentro del método `AddObjectInteraction`, se ejecuta una lógica diferente según la acción realizada:

- **Cuando el objeto es recogido:** Se crea una estructura de tipo `ObjectInteractionsStructure` y se rellenan los campos con los datos proporcionados como parámetros, incluyendo la hora actual como marca de inicio de interacción.
- **Cuando el objeto es soltado:** El sistema busca el objeto en el mapa que almacena los objetos con los que se ha interactuado, guardado como `Objects` en el `Level Structure` del nivel, para comprobar si ya existían interacciones previas:
 - * **Si el objeto ya estaba registrado:** Se recupera el arreglo de interacciones existente, se añade la nueva interacción (calculando el tiempo total que el objeto fue sostenido) y se actualiza en el mapa.
 - * **Si el objeto no estaba registrado:** Se crea una nueva entrada en el mapa con una lista que contiene únicamente la interacción actual.

En ambos casos, el cálculo de la duración de la interacción se realiza restando el tiempo inicial (registrado al recoger el objeto) al tiempo actual (registrado al soltarlo). Finalmente, los datos del objeto y sus interacciones actualizados se almacenan en el `Level Structure` del nivel actual.

```
{
  "TiempoAplicacion": 8,
  "Niveles": [
    {
      "Name": "L_Leve_Banyo",
      "HoraInicio": "4:51:21 PM",
      "Tiempo": 8,
      "Puzzles": [
        {
          "B_L_Patos": [
            {
              "HoraInicio": "4:51:22 PM",
              "Tiempo": 104.67800140380859
            }
          ],
          "Objects": [
            {
              "name": "Bote",
              "interactions": [
                {
                  "Hand": "Right",
                  "TimeOnPickUp": "4:51:55 PM",
                  "TimeOnDrop": "4:52:04 PM",
                  "Duration": 8
                }
              ]
            },
            {
              "name": "Llave_Patos",
              "interactions": [
                {
                  "Hand": "Left",
                  "TimeOnPickUp": "4:53:09 PM",
                  "TimeOnDrop": "4:53:10 PM",
                  "Duration": 0
                },
                {
                  "Hand": "Left",
                  "TimeOnPickUp": "4:53:10 PM",
                  "TimeOnDrop": "4:53:10 PM",
                  "Duration": 0
                }
              ]
            }
          ]
        }
      ],
      "Headset": [
        {
          "Timestamp": "4:51:24 PM",
          "Position": {
            "x": 34941.37890625,
            "y": -28591.724609375,
            "z": 2694.768310546875
          },
          "Orientation": {
            "x": 0.19244228303432465,
            "y": 0.97314971685409546,
            "z": -0.12627606093883514
          }
        }
      ]
    }
  ]
}
```

Figura 4.5: Ejemplo de resultado del fichero JSON generado.

5 Diseño del Escape Room

Este capítulo, en el que se muestra el proceso seguido para diseñar el *escape room*, se divide en seis apartados: la sección 5.1, introduce conceptos fundamentales sobre el diseño de un *escape room*; la sección 5.2, detalla el diseño de los niveles; la sección 5.3, aborda el diseño de los puzzles, describiendo la lógica y estructura de cada uno; la sección 5.4 se centra en el entorno, destacando su creación y personalización; la sección 5.5 donde se describen los elementos interactivos y decorativos diseñados; y finalmente, la sección 5.6, explora la incorporación de reminiscencia como herramienta cognitiva en el *escape room*.

5.1 Introducción

Existen muchos tipos de *escape rooms*, pero, por norma general, todos coinciden en que los puzzles son el centro de toda la experiencia.

La anatomía de un puzzle es compleja de definir, ya que no todos siguen la misma estructura; de lo contrario, acabarían volviéndose predecibles. Sin embargo, hablando específicamente de los puzzles diseñados para encontrarse dentro de un *escape room*, Wiemker, Elumir y Clare [36] definen los tres componentes básicos que debería tener cada uno de estos puzzles: un desafío, una solución y una recompensa.

El objetivo de los puzzles no debe ser engañar ni confundir al jugador. ‘*Como diseñador, quiero que las personas resuelvan mis acertijos. No me deleita el fracaso de las personas*’. — Errol Elumir (2018)[36]. Esto no significa que los acertijos deban ser fáciles, pero tampoco extremadamente difíciles: ‘*Your puzzles are meant to be solved*’. — Errol Elumir (2018)[36]. Por lo tanto, será necesario dedicar tiempo a ajustar su dificultad. Hay varias formas de hacerlo:

- **Pistas:** Cambiar la cantidad o claridad de la información para los jugadores.
- **Proximidad:** Cuanto más cerca esté la solución, más fácil será.
- **Guiar al jugador:** Dar pistas sutiles en la dirección correcta en respuesta a sus acciones.

Los acertijos no son interacciones independientes; forman parte del sistema completo de un escape room. Estas habitaciones siguen un patrón repetitivo de interacciones variadas que se conectan entre sí. Se busca garantizar que los jugadores no se pierdan y sepan qué hacer mediante la correcta implementación del ciclo de compulsión [20].

1. **Deseo:** Cada acertijo comienza con un deseo, como ir a algún lugar o conseguir algo.
2. **Obstáculo:** Este deseo se ve bloqueado por un obstáculo, que es el acertijo en sí.

3. **Recompensa:** Al superar el obstáculo, los jugadores reciben una recompensa que apunta hacia el siguiente deseo.

4. **Repetir:** Este ciclo se repite, guiando a los jugadores de un desafío al siguiente.

Repetir este ciclo lleva a los jugadores de un acertijo al siguiente. Cuando esto no está claro, los jugadores se pierden y pierden interés. Mantener el ciclo de compulsión es esencial para que los jugadores disfruten de la escape room en su conjunto.

5.2 Diseño del nivel

La distribución y colocación de los acertijos desempeñan un papel importante en cualquier escape room. El bucle de juego rara vez es completamente lineal y, en la mayoría de los casos, puede parecer incluso caótico. Sin embargo, los enfoques más comunes suelen incorporar el ciclo de compulsión de alguna manera. Heikkinen y Shumeyko (2016)[13] identifican tres modelos principales:

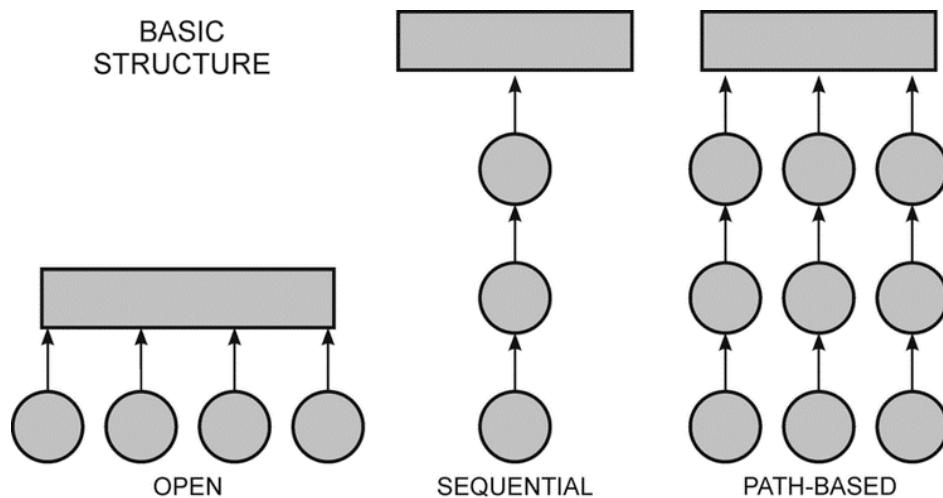


Figura 5.1: Tipos de organización de los puzzles

1. **Abierto:** Todos los acertijos son independientes y no dependen entre sí para escapar de la sala.

- **Pros:** Facilita la resolución paralela de acertijos, lo que permite que todos trabajen individualmente.
- **Contras:** Experiencia individualista. Es difícil controlar la progresión de la dificultad.

2. **Secuencial:** Una experiencia mayormente (o completamente) lineal. Cada acertijo lleva a uno nuevo.

- **Pros:** Fácil de seguir y ofrece un fuerte efecto del ciclo de compulsión. Mayor control sobre la progresión de la dificultad.

- **Contras:** No permite la paralelización. Todo el equipo debe trabajar en un solo desafío a la vez, lo que puede volverse tedioso o aburrido.
3. **Basado en rutas:** Hay múltiples rutas de acertijos disponibles al mismo tiempo, que generalmente conducen a un único acertijo final (meta-acertijo).
- **Pros:** Permite trabajo en equipo y paralelización, lo que favorece grupos más grandes. Si el meta-acertijo indica qué rutas no se han completado, los jugadores siempre saben qué hacer.
 - **Contras:** Puede distraer o confundir a los jugadores fácilmente.

En el proyecto se usará una combinación de diseño de escape room de tipo abierto y secuencial, donde predominará el enfoque abierto, con la excepción de algunos puzzles donde la solución de este puzzle se encontrará tras resolver un acertijo previo. Esto cuadra con las ventajas y desventajas de cada tipo, especialmente al tratarse de un escape room diseñado para un único jugador.

Cada nivel contará con una serie de puzzles que, al ser completados, entregarán al jugador una llave como recompensa. Estas llaves deberán colocarse en cerraduras ubicadas a los lados de la puerta principal de la sala. Cuando todas las cerraduras contengan una llave, el nivel contará como completado y se abrirá la puerta, permitiendo al jugador salir del nivel. Una vez complete el nivel, volverá a una zona neutra, la misma sala donde aparece al inicio del juego. En este momento podrá elegir otro nivel al que adentrarse y resolverlo si así lo desea. Estos niveles ayudarán a clasificar según el profesional vea con qué tiene dificultades, ya que cada nivel cuenta con duplicas donde la diferencia es la dificultad de los puzzles que contiene.

5.3 Diseño de puzzles

A la hora de diseñar un escape room, es necesario prestar especial atención para crear puzzles justos y entretenidos. Errol Elumir [10] explica que los diseñadores de *escape rooms* deben tratar de diseñar puzzles que puedan resolverse únicamente con la información contenida en el nivel. Elumir explica que existen unas reglas básicas para el diseño de los puzzles de un escape room. entre ellas encontramos:

Un puzzle debe tener una única solución: Los jugadores tienden a irse por caminos equivocados: encontrarán una teoría incorrecta y la expandirán hasta llegar tan lejos por el camino equivocado que podrían estar jugando un juego diferente. Sin embargo, si las pistas los guían en esa dirección, no es realmente culpa de ellos.

Las pistas deben estar claramente vinculadas a los rompecabezas: Las conexiones entre las pistas y los rompecabezas deben ser naturales y temáticas, evitando confusiones o caos durante la resolución.

Los rompecabezas deben ser breves: Deben resolverse en un máximo de cinco minutos para mantener el flujo del juego. Puzzles largos o tediosos pueden arruinar la experiencia.

Evitar estados irrecuperables: Un rompecabezas no debe depender de elementos que puedan moverse, perderse o destruirse, ya que esto puede hacer que el juego sea imposible de completar.

Feedback inmediato: El sistema debe comunicar claramente si una solución es correcta, mejorando la satisfacción del jugador y reduciendo la necesidad de pedir pistas. En nuestro caso, esto se logra mediante sonidos al completar el puzzle, además de la recompensa principal.

Consistencia en patrones: Las reglas y patrones del juego deben ser coherentes para evitar confundir o frustrar a los jugadores. Cambios inesperados en los patrones rompen la confianza del jugador en el diseño. Para garantizar esta consistencia, hemos mantenido la misma mecánica principal a lo largo de todo el juego: la colocación de objetos.

Testeo y ajustes constantes: Los diseñadores suelen subestimar la dificultad de sus rompecabezas. Probarlos con jugadores novatos y expertos asegura que sean divertidos, claros y alcanzables.

Sabemos ya que la gamificación promueve un envejecimiento saludable. Usaremos todos estos principios para diseñar puzzles que fomenten este envejecimiento saludable y frenen el deterioro cognitivo, fomentando la reserva cognitiva, que es la encargada de preservar nuestras facultades mentales.

Recordamos que el deterioro cognitivo puede afectar a diversas capacidades cognitivas, como puede ser la memoria, la concentración y la capacidad de mantenerla, el procesamiento de la información o las habilidades lingüísticas, dificultando de esta forma a la habilidad de una persona de comunicarse por no encontrar las palabras deseadas.

El objetivo principal de este proyecto es fomentar la estimulación cognitiva a través de un juego. Es importante señalar que, para que la estimulación cognitiva sea efectiva, es importante que se adapte al nivel cognitivo y potencial de cada persona. Por ello, se han diseñado diferentes puzzles que cuentan con dos y hasta tres variaciones según el nivel de dificultad seleccionado. Entendiendo que el cerebro y sus capacidades cognitivas funcionan como una red, es decir, todas las capacidades funcionan de modo conjunto; por ejemplo, para mejorar la memoria o el lenguaje, es importante trabajar también la atención y el razonamiento. Por este motivo, usualmente, un mismo ejercicio entrenará más de una habilidad cognitiva, y a su vez, una misma habilidad podrá ser trabajada con diferentes ejercicios.

El Escape room contará con puzzles y actividades para trabajar las distintas capacidades cognitivas:

- **Ejercicios para la comprensión y percepción:** La percepción es uno de los procesos cognitivos, capaz de captar, procesar y dar sentido de forma activa a la información que alcanza nuestros sentidos. Por otro lado, llamamos comprensión a la capacidad que nos permite entender esta información que hayamos percibido. La comprensión es una de las capacidades cognitivas básicas que se encuentra íntimamente relacionadas con otras habilidades, de forma que ejercitárla y desarrollarla puede mejorar el rendimiento de esta y otras funciones cognitivas.
- **Ejercicios para el lenguaje:** El lenguaje es una facultad cognitiva que tenemos para expresar pensamientos y sentimientos a través de la palabra. Es fundamental para que las personas puedan comunicarse con otras y, por tanto, es importante trabajarla.
- **Ejercicios para el razonamiento:** Se entiende por razonamiento a la facultad que permite resolver problemas, extraer conclusiones y aprender de manera consciente de

los hechos, estableciendo conexiones causales y lógicas entre ellos. El razonamiento es una de las funciones cognitivas superiores que nos permite pensar y tomar decisiones ante los estímulos, eventos y situaciones donde podamos encontrarnos.

- **Ejercicios para las gnosias:** Las gnosias son la habilidad que tiene el cerebro para reconocer información previamente aprendida como pueden ser objetos, personas o lugares que captamos a través de nuestros sentidos. Dado que las gnosias nos ayudan a identificar y reconocer información que ya hemos aprendido previamente, es importante entrenarlas para no invertir más recursos, como puede ser el tiempo, de los necesarios en su reconocimiento.
 - **Ejercicios para el control ejecutivo:** Las funciones ejecutivas se pueden definir como el conjunto de capacidades cognitivas necesarias para controlar y autorregular la propia conducta. Estas funciones son actividades mentales complejas necesarias para planificar, organizar, guiar, etc. El control ejecutivo es una de las capacidades cognitivas esenciales, está encargada de la regulación de otras funciones.
 - **Ejercicios para la planificación:** La planificación es una habilidad cognitiva fundamental que forma parte de las funciones ejecutivas. La planificación se puede definir como la habilidad de “pensar en el futuro”, de anticipar mentalmente la forma correcta de ejecutar una tarea o alcanzar una meta específica.
 - **Ejercicios para el cálculo:** El cálculo es una facultad mental que, si no se ejercita, pierde agilidad y fluidez. Se trabaja realizando operaciones matemáticas mentalmente sin recurrir a herramientas como calculadoras.
 - **Ejercicios para la atención:** La atención hace referencia al estado de observación y alerta que permite tomar conciencia de lo que ocurre en el entorno. Las actividades que la trabajan tienen como objetivo potenciar y controlar diferentes tipos de atención, como la atención selectiva.
 - **Ejercicios para la memoria:** La memoria es una de las capacidades cognitivas que primero empiezan a deteriorarse como consecuencia de la edad, para contrarrestar este declive es importante mantener la mente activa. Además de trabajar la memoria en general se trabajarán distintos tipos:
 - **La memoria semántica**, incluye a todos aquellos conocimientos generales que no implican el recuerdo de sucesos concretos.
 - **La memoria procedural**, es responsable de las habilidades y hábitos que realizamos de forma automática.
 - **La memoria de trabajo**, se refiere a la capacidad que nos permite mantener en la mente los elementos que necesitamos para realizar una tarea mientras la estamos ejecutando.
 - **Ejercicios para las praxias visoconstructivas:** Las praxias visoconstructivas son las facultades mentales encargadas de planificar y realizar los movimientos necesarios para organizar una serie de elementos en el espacio para formar un dibujo o figura final.
-

- **Ejercicios para la velocidad de procesamiento:** La velocidad de procesamiento es un proceso mental que establece la relación entre la ejecución cognitiva y el tiempo invertido. Ejercitárla permitirá procesar la información de manera más rápida sin perder su eficiencia y mejorándola.
- **Ejercicios para la abstracción:** La abstracción es la capacidad cognitiva de extraer la esencia o las características fundamentales de una información o fenómeno complejo, ignorando los detalles irrelevantes o superficiales.

5.3.1 Puzzles

Teniendo en cuenta todo lo anterior, a continuación, se muestran todos los puzzles separados por niveles en los que se encuentran. Se explica cómo se resuelve cada uno y se exponen las diferencias entre aquellos que cuentan con distintos niveles de dificultad.

COCINA			
Puzzle Receta			
Área trabajada	Elementos	Dificultad	Solución
Atención selectiva y memoria semántica	Libro de recetas, Paellera, Ingrediente	Difícil	La clave de este puzzle se encuentra en el libro de recetas, donde la página abierta contendrá los pasos a seguir para realizar una paella de marisco. En este texto se puede leer la cantidad de cada tipo de ingrediente que el jugador deberá colocar dentro de la paellera para completar el puzzle.
		Fácil	En la versión fácil de este puzzle la receta muestra directamente en la sección de ingredientes una lista de estos.

Tabla 5.1: Detalles del puzzle receta



(a) Receta dificultad difícil

(b) Receta dificultad fácil

Figura 5.2: Aspecto final del puzzle receta del nivel cocina (Figura 5.17)

COCINA			
Puzzle Torre de Magdalenas			
Área trabajada	Elementos	Dificultad	Solución
Razonamiento y comprensión	Torre pastelera de tres niveles de magdalenas, Candado numérico	Difícil	Este puzzle cuenta con un candado numérico que se soluciona al introducir una combinación de tres números específica. La combinación correcta es la cantidad de magdalenas que hay en cada nivel de la torre. El candado está decorado con una etiqueta. Esta etiqueta contiene un dibujo que hace referencia a que la combinación correcta es el número de magdalenas que hay en cada nivel de la torre.
		Fácil	Al bajar la dificultad del nivel, en este puzzle el dibujo de la etiqueta es una pista más obvia.

Tabla 5.2: Detalles del puzzle torre de magdalenas



(a) Etiqueta versión difícil (b) Torre de magdalenas (c) Etiqueta versión fácil

Figura 5.3: Aspecto final del puzzle torre de magdalenas del nivel cocina (Figura 5.17)

COCINA			
Puzle Poner la Mesa			
Área trabajada	Elementos	Dificultad	Solución
Percepción, comprensión y atención	Platos, Tenedores, Cuchillos, Mesa comedor	Global	En medio de la sala se encuentra una mesa a mitad de poner. Al rededor de la mesa hay colocadas 4 sillas y enfrente de cada una de ellas se encuentra un conjunto incompleto de vajilla. El conjunto completo estaría formado por un plato, un cuchillo y un tenedor, pero en tres de las posiciones falta uno de los elementos. El objetivo será buscar por la sala los elementos que falta en cada lugar y colocarlos para completar la vajilla de toda la mesa.

Tabla 5.3: Detalles del puzzle de poner la mesa



Figura 5.4: Aspecto final del puzzle poner la mesa dentro del nivel cocina (Figura 5.17)

BAÑO			
Puzzle Patos de Goma			
Área trabajada	Elementos	Dificultad	Solución
Gnosia visual, control atencional, razonamiento y velocidad de procesamiento	Patos de bañera de distintos colores, candado numérico	Difícil	Este puzzle se resolverá en el momento se introduzca en el candado la combinación de números correcta. Cada número del candado está asociado a un color. Dentro de la bañera se pueden observar varios patos de goma de cada color. El jugador deberá contar cuantos patos de cada color hay e introducirlo en el número del candado correspondiente a dicho color. En esta dificultad hay patos animados moviéndose por la bañera complicando su seguimiento.
		Media	En esta dificultad los patos ya no se encontrarán en movimiento. Sin embargo el jugador deberá prestar atención puesto que no todos los patos se encontrarán dentro de la bañera.
		Fácil	En la versión más fácil del puzzle los patos estarán estáticos todos dentro de la bañera.

Tabla 5.4: Detalles del puzzle patos de goma del nivel baño



Figura 5.5: Aspecto final del puzzle patos de goma

BAÑO			
Puzzle Palabras Encadenadas			
Área trabajada	Elementos	Dificultad	Solución
Control atencional, memoria y fluidez verbal	Chocolate, Teléfono, Nocilla, Lámpara, Radio, Pedestales	Difícil	En este puzzle se encuentran cinco pedestales en fila. El puzzle se completa cuando en cada pedestal se coloca el objeto correcto. El objeto correcto de cada pedestal se determina siguiendo las reglas del juego tradicional “palabras encadenadas” de forma que la última sílaba del nombre de un objeto debe coincidir con la primera sílaba del nombre del objeto colocado a su derecha. Por lo tanto, el orden correcto será Chocolate - Teléfono - Nocilla - Lámpara - Radio
		Fácil	En la versión más fácil de este puzzle cada pedestal incluye el nombre del objeto que va sobre él.

Tabla 5.5: Detalles del puzzle palabras encadenadas



Figura 5.6: Aspecto final del puzzle de palabras encadenadas del nivel baño (Figura 5.15) en su versión fácil

BAÑO			
Puzle Secuencia Botes de Ducha			
Área trabajada	Elementos	Dificultad	Solución
Praxia visoconstructiva, razonamiento abstracto, control ejecutivo y control atencional	Una serie de botes de ducha de distintos colores y tamaños	Difícil	El puzle se completará cuando los botes estén ordenados. El criterio de ordenación para la dificultad difícil es que la figura que forman las etiquetas de los botes sea una diagonal. Sin embargo, el jugador deberá llevar cuidado y no dejarse engañar y tratar de ordenarlos por color o tamaño de los botes. El jugador si presta atención a los botes fijos que no puede cambiar de posición verá que su única opción de conseguir un orden serán las etiquetas.
		Media	Al bajar la dificultad a media, se añade un nuevo criterio para ordenar los botes, ya no solo se ordenan por la etiqueta, sino también por altura de los botes.
		Fácil	En la versión fácil del puzle el usuario puede seguir varios criterios de ordenación para conseguir el mismo resultado, ordenar por tamaño, por colores y por altura de las etiquetas.

Tabla 5.6: Detalles del puzzle secuencia de botes



(a) Versión fácil del puzzle Secuencia de botes resuelto



(b) Versión media del puzzle Secuencia de botes resuelto



(c) Versión difícil del puzzle Secuencia de botes resuelto

Figura 5.7: Aspecto final del puzzle secuencia de botes del nivel baño (Figura 5.15)

BAÑO			
Puzle Planificación para Ducharse			
Área trabajada	Elementos	Dificultad	Solución
Funciones ejecutivas y planificación	Tablero, Tarjetas de Acción	Global	El puzzle cuenta con seis tarjetas, cada una contiene un texto describiendo una acción entre “Quitarse la ropa”, “Entrar a la ducha”, “Lavarse”, “Apagar el agua”, “Secarse” y “Vestirse”. En el tablero se le pide al jugador que coloque las tarjetas ordenando estas acciones para cumplir los pasos necesarios para ducharse. El puzzle se resolverá cuando las acciones sigan el orden lógico, por ejemplo, no entrar a la ducha todavía con ropa puesta.

Tabla 5.7: Detalles del puzzle planificación de ducharse

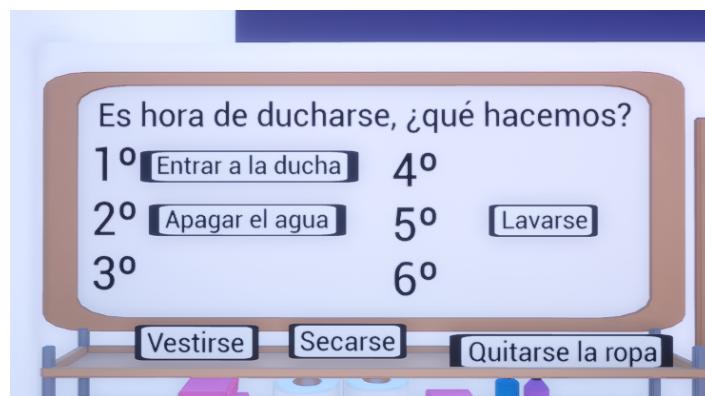


Figura 5.8: Aspecto final del puzzle de planificación para ducharse del nivel baño (Figura 5.15) sin resolver

SALÓN			
Puzle Figuras 3D			
Área trabajada	Elementos	Dificultad	Solución
Gnosia visual y abstracción	Conjuntos de figuras geométricas básicas de colores, Pedestales, Cuadros	Difícil	<p>La base de este puzzle es una zona con cuatro cuadros alineados con pedestales. Cada pedestal espera que se coloque una figura sobre él. Cada cuadro contiene una imagen representando como se ve desde arriba uno de los conjunto de figuras. Para completar el puzzle el jugador deberá colocar cada figura en el pedestal correspondiente al cuadro que la represente. El jugador deberá fijarse y prestar atención tanto a las formas como a los colores de las formas geométricas puesto que en esta dificultad ninguno de los dos se mantiene, una misma figura geométrica puede tener disenso colores según el conjunto en le que se encuentre, además, deberá prestar atención a la posición con respecto al resto de figuras geométricas del conjunto será determinante.</p>
		Media	<p>Para facilitar la resolución del puzzle en esta dificultad las figuras geométricas no mantienen la misma posición ni orientación pero sí sus colores, es decir, si la esfera es roja en uno de los conjuntos lo será también en los demás. De esta forma la atención se puede centrar toda en la distribución de las figuras.</p>
		Fácil	<p>En la dificultad fácil las formas geométricas no solo mantienen en todos los conjuntos el mismo color, pero además, no hay un conjunto que contenga las mismas figuras ,de esta forma el jugador podrá resolverlo fijándose tanto en los colores como en las formas, ya que no habrá dos conjuntos con la misma combinación de colores ni de formas.</p>

Tabla 5.8: Detalles del puzzle vistas de figuras

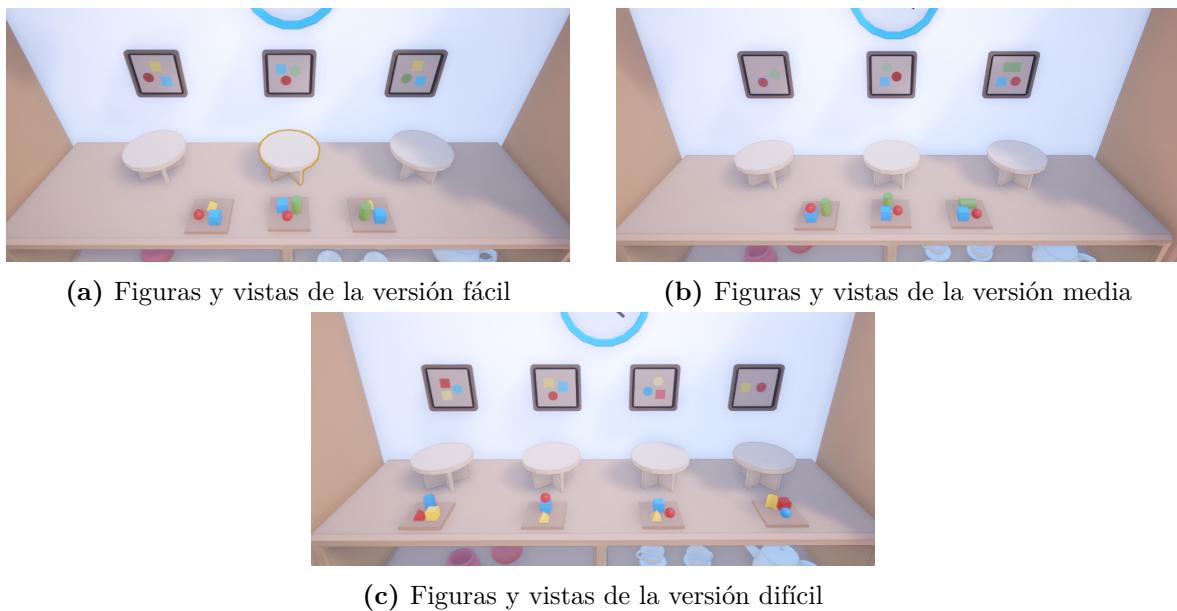


Figura 5.9: Aspecto final del puzzle figuras 3D del nivel salón (Figura 5.16)

SALÓN			
Puzzle Refraneas			
Área trabajada	Elementos	Dificultad	Solución
Lenguaje, memoria, razonamiento y abstracción	Tablero, Tarjetas con texto	Global	<p>Este puzzle está situado dentro de un tablero vertical colocado en una pared de la sala. Dentro de este tablero se puede leer una instrucción que indica que se ha de completar los refranes, debajo se observan la primera mitad de cuatro refranes. El puzzle cuenta con cuatro tarjetas interactivas, cada una contiene el final de un refrán. La misión del jugador será relacionar cada inicio de refrán con su final y colocar entonces la tarjeta en la posición correcta. Los refranes completos partidos de la misma forma que en el juego son: de tal palo / tal astilla, a caballo regalado / no le mires el dentado, a quien madruga / Dios le ayuda y perro ladrador / poco mordedor.</p>

Tabla 5.9: Detalles del puzzle de refranes

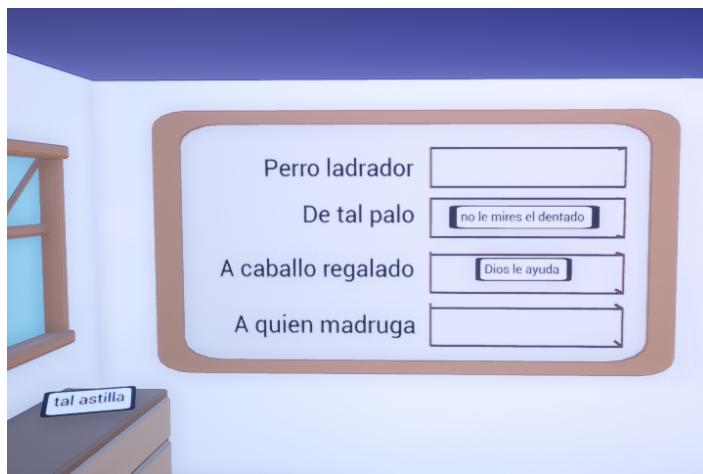


Figura 5.10: Aspecto final del puzzle conexión de refranes del nivel salón (Figura 5.16)

SALÓN			
Puzle Colección de Libros			
Área trabajada	Elementos	Dificultad	Solución
Praxia visocronstructiva y razonamiento	Colección de libros, libros sueltos, candado numérico	Difícil	<p>Dentro de una de las estanterías del salón se podrá observar una colección de libros cuyos lomos al estar colocados correctamente forman un dibujo. Sin embargo faltarán diversos tomos para que el dibujo este completo. En los libros sueltos están vinculados a la colección ya que su lomo claramente mantiene el mismo patrón. Los libros que faltan estarán dispersos por todo el nivel y el jugador deberá encontrarlos y colocarlos en su posición. Una vez la colección este al completo en el dibujo que se forma se podrá observar un código de 4 números. Este código será la solución al candado numérico del nivel. El jugador deberá introducir el código para completar el puzle.</p>
		Fácil	<p>Para facilitar la resolución de este puzle, en la dificultad más fácil faltarán menos libros en la colección. Además, los libros sueltos se encontrarán más a la vista y más cerca de la estantería. Otro detalle será que no faltarán dos libros consecutivos en la colección para facilitarle al jugador ver el patrón.</p>

Tabla 5.10: Detalles del puzzle colección de libros

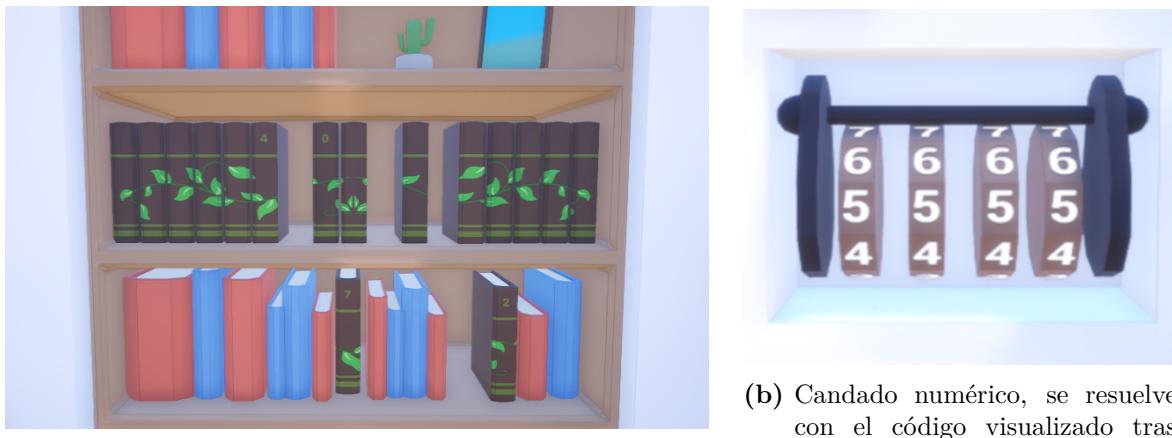


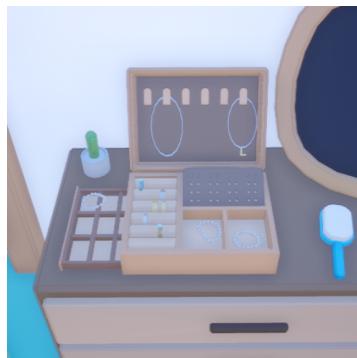
Figura 5.11: Aspecto final del puzzle colección de libros del nivel salón (Figura 5.16)

ALCOBA			
Puzzle Joyero			
Área trabajada	Elementos	Dificultad	Solución
Control atencional y razonamiento	Joyería, joyas (collares, anillos, pulseras), candado numérico	Global	Este puzzle cuenta con un candado numérico que se soluciona al introducir una combinación de tres números específica. La combinación correcta es la cantidad de cada tipo de joya que se encuentra dentro del joyero que se puede encontrar en la sala. El candado está decorado con una etiqueta. Esta etiqueta contiene un dibujo que hace referencia a que la combinación correcta es la cantidad de collares, anillos y pulseras que hay en total dispersado por el joyero.

Tabla 5.11: Detalles del puzzle del joyero y sus joyas



(a) Etiqueta versión fácil



(b) Joyero y joyas



(c) Etiqueta versión difícil

Figura 5.12: Aspecto final del puzzle del joyero del nivel alcoba (Figura 5.18)

ALCOBA			
Puzzle Vestirse			
Área trabajada	Elementos	Dificultad	Solución
memoria procedimental, memoria semántica y planificación	Maniquí, Tarjetas con imágenes de prendas de ropa, Cuadro con contexto	Difícil	<p>El puzzle se centra en un maniquí que se encuentra entre dos armarios. Dentro de estos armarios a su lado habrán tarjetas con imágenes de prendas de ropa. La actividad consiste en vestir al maniquí de forma adecuada. En la versión difícil del puzzle el puzzle cuenta con un cuadro que añade un contexto específico al acertijo. El cuadro explica un plan salir al exterior de la casa, muestra que nos encontramos a mitades de enero y que se espera un clima frio en la calle. Con esta información el jugador deberá entender que debe vestir al maniquí con prendas de abrigo. En esta dificultad, también habrán más opciones de prendas de ropa entre las que elegir, dificultando acertar aleatoriamente.</p>
		Fácil	<p>En la dificultad fácil, el jugador para completar el puzzle bastará únicamente con vestir por completo al maniquí de forma lógicas, es decir colocar zapatos en los pies y no en otras partes del cuerpo, igual con unos pantalones en las piernas, una camiseta en el torso y un sombrero en la cabeza.</p>

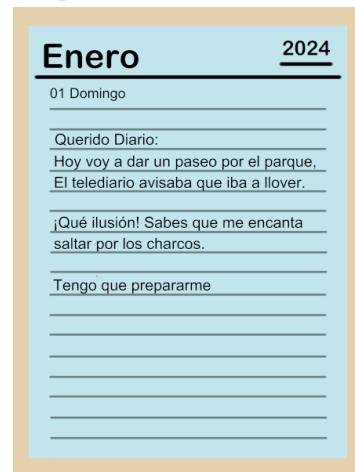
Tabla 5.12: Detalles del puzzle de vestirse



(a) Zona central del puzzle



(b) En la dificultad difícil: diario donde se encontrará la pista sobre que va a llover y por lo tanto hay que escoger la ropa de lluvia



(c) Contenido del diario

Figura 5.13: Aspecto final del puzzle del vestirse del nivel alcoba (Figura 5.18)

TODOS LOS NIVELES			
Puzle Puerta Final			
Previo	Elementos	Dificultad	Solución
Para resolverlo será necesario previamente haber resuelto todos los puzzles restantes del nivel	Llaves recompensa, Cerraduras, Puertas	Global	Al completar cada uno de los puzzles del nivel el jugador recibe una llave como recompensa. Al lado de la puerta de salida del nivel se encuentran tantas cerraduras como acertijos haya en la sala, el jugador deberá colocar una llave en cada una de las cerraduras para completar el puzzle y lograr que la puerta se abra. Una vez hecho, se habrá completando el nivel y se le permitirá al jugador salir de él.

Tabla 5.13: Detalles del puzzle final de cada nivel



(a) Estándar puerta final con candados



(b) Ejemplo de llave que se consigue tras completar un puzzle dentro de un nivel. Las llaves se colocarán en una cerradura

Figura 5.14: Aspecto de la puerta para salir del nivel alcoba

5.4 Entorno

Como explica Rachel Sugar en su artículo [31], uno de los principales objetivos de los *escape rooms* es sumergir al jugador en otro mundo, donde se olviden momentáneamente de su realidad. Este tipo de experiencias altamente inmersivas provocan una respuesta emocional en los usuarios, un aspecto que se analizará más adelante cuando tratemos la reminiscencia.

Los *escape rooms* cuentan con entornos cuidadosamente diseñados, donde cada objeto tiene un propósito específico que aporta a la experiencia, ya sea por ser parte de un puzzle, de su contexto o para añadir coherencia al entorno. Para optimizar la experiencia, resulta más efectivo usar un diseño minimalista frente a uno maximalista, dado que un espacio sobre-saturado de objetos provocaría que el jugador dedicase más tiempo a identificar los acertijos que en resolverlos. Por estos motivos se ha buscado encontrar un equilibrio entre un entorno claro y definido sin sentirse medio vacío.

A la hora de crear el entorno, otro factor realmente importante en nuestro caso es tener en cuenta a nuestro público objetivo. Es posible que sea la primera vez del usuario entrando a un mundo virtual. Buscamos evitar una desorientación espacial o temporal, por lo que se tomó la decisión de ambientar el escape room en un ambiente no desconocido para ellos: una casa que pudieran reconocer y que les recordara a su propio hogar.

Cada habitación será un nivel independiente del escape room y representará una de las estancias principales de un hogar: la cocina, el salón, el baño y la alcoba. Además, se ha diseñado un recibidor, que será donde el jugador inicie su aventura y elija el nivel que desea intentar resolver. Se han diseñado las estancias de forma que transmitan una sensación de amplitud. Se busca aportarle al jugador libertad total, tanto a la hora de la resolución de los puzzles del nivel, como también en cuanto a poder explorar y moverse por el entorno sin restricciones.

Para construir la estructura base de las habitaciones, se ha utilizado un paquete de *assets* externos¹ que incluía la mayoría de los elementos necesarios para el diseño de la parte estática de los niveles. Finalmente, se ha obtenido el siguiente resultado:

¹<https://www.fab.com/listings/462bde75-b156-4ba9-8276-b4408cbc5796>

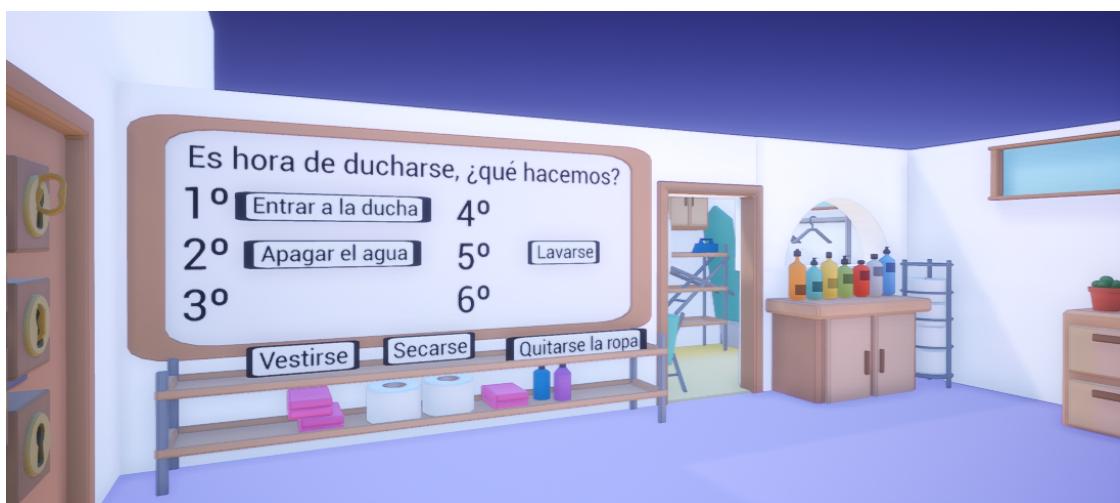


Figura 5.15: Diseño final de la estancia baño desde distintos ángulos



Figura 5.16: Diseño final de la estancia salón desde distintos ángulos



Figura 5.17: Diseño final de la estancia cocina desde distintos ángulos



Figura 5.18: Diseño final de la estancia alcoba desde distintos ángulos



Figura 5.19: Diseño final de la estancia **recibidor** desde distintos ángulos

5.5 Elementos

Para poder personalizar completamente los puzzles conforme se habían ideado, se ha modelado y texturizado la mayoría de los componentes y objetos interactivos específicamente para este proyecto. Esto se ha logrado empleando la herramienta Blender para todos sus pasos: modelado 3D, *UV mapping*, texturizado y personalización de las colisiones. A continuación, se explica brevemente en qué consiste cada una de estas etapas.

- El **modelado 3D** consiste en crear representaciones tridimensionales de objetos mediante software especializado. Permite crear desde personajes y entornos hasta objetos y superficies. A través de este proceso, se construyen formas geométricas conocidas como polígonos, que al combinarse forman la estructura del modelo. Tras este proceso se obtiene como resultado una malla tridimensional la cual carece de estilo visual.
 - El **texturizado en 3D** consiste en añadir texturas a la superficie de un objeto 3D para definir su apariencia visual. Esto incluye la creación de texturas (a partir de fotos o de cero). Para lograr una coherencia estética, los elementos se han diseñado manteniendo el estilo visual y empleando la misma paleta de colores que los elementos base de la habitación. Para texturizar se ha empleado la técnica del mapeo UV (*Uv mapping*). Este método permite desplegar la superficie tridimensional de los objetos en un plano bidimensional, facilitando la aplicación de texturas.

A continuación, se muestran todos estos elementos propios:



Figura 5.20: Colección de objetos propios

5.6 Reminiscencia

La terapia de reminiscencia consiste en rememorar eventos de la historia personal del sujeto mediante la estimulación de recuerdos con significado personal y que evoquen sentimientos positivos [17]. Esto se logra mediante estímulos como fotografías, música y objetos cotidianos.

El objetivo principal de la terapia de reminiscencia [14] es la estimulación de la memoria episódica autobiográfica, pero además otorga una estimulación cognitiva colateral. Con esta terapia también se estimulan campos como la atención, el lenguaje expresivo y comprensivo, la memoria semántica y las gnosias, entre otros aspectos.

La terapia de reminiscencia, puede tener efectos positivos en personas con demencia, ayudándolas a mejorar en áreas como su calidad de vida, habilidades cognitivas, comunicación y estado de ánimo. Un estudio [37] realizado en residencias de ancianos muestra una mayor variedad de beneficios, incluyendo mejoras en la calidad de vida, la memoria y la capacidad de comunicarse, especialmente cuando la terapia se realiza de forma continua. Además, el mismo estudio sugiere que, cuando la terapia de reminiscencia se realiza de forma individual, parece tener un impacto positivo tanto en el estado de ánimo como en la función cognitiva de quienes participan.

Se puede destacar la capacidad de la realidad virtual de generar un entorno inmersivo que facilita la evocación de recuerdos, mejorando y enriqueciendo la experiencia de reminiscencia.

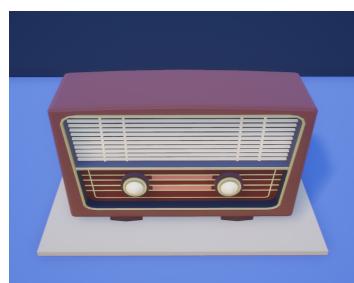
En este proyecto, se han diseñado elementos específicos para incorporar la terapia de reminiscencia dentro del escape room virtual. Estos objetos cumplen una doble función: ser elementos interactivos de los puzzles del juego y catalizar o fomentar recuerdos significativos para los usuarios.

Los elementos son los siguientes:

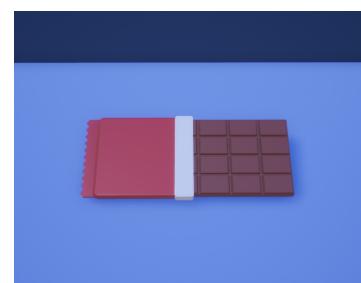
- Un teléfono que imita el aspecto que tenían los teléfonos antiguos. En la década de 1950 comenzaron a ser comunes, y a inicios de la década de 1960 se popularizaron en los hogares. (Figura 5.21a)
- Una antigua radio: entre las décadas de 1920 y 1950, la radio fue el medio de entretenimiento familiar dominante. (Figura 5.21b)
- Una tableta de chocolate que busca evocar el recuerdo de la merienda tradicional de “una pastilla de chocolate y un currusco de pan”. (Figura 5.21c)



(a) Teléfono antiguo



(b) Radio antigua



(c) Tableta de chocolate

Figura 5.21: Elementos diseñados para la terapia de reminiscencia

6 Conclusiones

En este capítulo se valora el trabajo realizado. En la sección 6.1 se extraen las conclusiones del proyecto, mientras que las futuras líneas de trabajo se presentan en la sección 6.2.

6.1 Conclusiones

Este proyecto se ha dividido en cuatro partes: investigación acerca del deterioro cognitivo y técnicas para su prevención y frenado; capitalización de la tecnología de la que se partía y programación de un sistema de puzzles y las mecánicas que se emplearán; diseño y desarrollo del propio *escape room* y sus puzzles; y, por último, la generación de un sistema de seguimiento para monitorizar al jugador.

En la primera parte, se descubrió qué actividades son eficaces ante el deterioro cognitivo, lo importante que es fomentar la estimulación cognitiva y trabajar distintas capacidades cognitivas de forma habitual para prevenirlo. En este momento empezaron a surgir ideas de puzzles y se empezó a dar forma a las mecánicas que habría que desarrollar. En la siguiente etapa, se terminó por especificar que la mecánica principal sería colocar objetos, y se procedió a programar todo lo necesario para el funcionamiento del juego y la ejecución del sistema de seguimiento. Tras terminar la programación, llegó el momento de diseñar definitivamente los distintos puzzles y niveles que tendría el *escape room*. Se procedió entonces a crear todos los elementos necesarios y a añadirlos al proyecto.

El desarrollo de este escape room en realidad virtual ha demostrado la gran cantidad de aplicaciones que puede tener esta tecnología, muchas de las cuales aún están por descubrir. La integración de *hand tracking* elimina barreras tecnológicas y ofrece una experiencia inmersiva sin necesidad de controladores adicionales. La gamificación y el diseño de puzzles basados en reminiscencias se consolidan como estrategias eficaces para mantener el interés y la motivación de los usuarios, mientras que la variedad de entornos y desafíos propuestos fomenta un entrenamiento integral de diversas capacidades cognitivas.

Al terminar este documento y echando la vista atrás, se puede observar que se han conseguido cumplir todos los objetivos marcados al inicio. Sin embargo, mientras se desarrollaba todo el proyecto, no he podido evitar pensar en más elementos que añadirían valor al proyecto, de los cuales entraré en detalle en la próxima sección 6.2.

Por último, me gustaría añadir un par de palabras más personales. No puedo evitar sentir una gran sensación de orgullo al observar el resultado final de tanto esfuerzo. Estoy agradecida con este trabajo, me ha permitido superar nuevos retos, además de otorgarme la oportunidad de aprender a manejarlo y a programar en uno de los motores más empleados en la industria

del entretenimiento digital.

6.2 Trabajo futuro

A futuro, sería interesante ampliar el proyecto desarrollando un sistema que, utilizando los datos almacenados en los archivos JSON generados, permita recrear visualmente el avance del usuario por el nivel. Donde se pudiese observar los pasos del jugador, su orientación, las interacciones con los elementos el entorno y el proceso de resolver un puzzle. De esta forma facilitaría el análisis del desempeño del jugador, se podría observar de forma más clara las áreas donde el usuario presentase mayor dificultas o la rapidez tiene al resolver los acertijos, la velocidad de toma de decisiones y su capacidad de adaptarse a nuevas tecnologías.

Otra posible línea de trabajo sería crear la capacidad de puzzles adaptativos. Esto aseguraría que, una vez resuelto un desafío, las siguientes iteraciones presentasen variaciones, evitando que los usuarios pierdan interés y fomentando un esfuerzo mental sostenido. Estas variaciones podrían incluir cambios en las pistas, resultados, o la mecánica de los puzzles.

Además, crear más contenido, se podrían desarrollar nuevos niveles y entornos temáticos que explorasen nuevas capacidades cognitivas o puzzles que trabajasen las mismas pero de distintas nuevas formas.

Por último, podría ser interesante preparar el proyecto para en entornos clínicos y residenciales, añadiendo funcionalidades específicas para terapeutas y cuidadores. Por ejemplo, incluir paneles de control que permitan controlar el flujo del juego, monitorizar el progreso de los usuarios, personalizar sus objetivos o generar informes detallados.

Bibliografía

- [1] Materiales instanciados. <https://dev.epicgames.com/documentation/es-mx/unreal-engine/instanced-materials-in-unreal-engine>.
- [2] J. Albartus. Mystery of time and space. <https://en.wikipedia.org/wiki/MOTAS>, 2001.
- [3] J. N. Bailenson, N. Yee, J. Blascovich, A. C. Beall, N. Lundblad, and M. Jin. The use of immersive virtual reality in the learning sciences: Digital transformations of teachers, students, and social context. *Journal of the Learning Sciences*, 17(1):102–141, 2008.
- [4] M. Berkman. History of virtual reality. *Encyclopedia of Computer Graphics and Games*, 2024.
- [5] P. Brunet and A. Vinacua. Introducción a la realidad virtual. <https://www.cs.upc.edu/~virtual/SGI/guiions/ArquitecturaRV.pdf>, 2024.
- [6] P. R. Canfranc. El crecimiento silencioso de la realidad virtual. 2020.
- [7] W. Crowther and D. Woods. Advent, colossal cave o adventure. https://es.wikipedia.org/wiki/Colossal_Cave_Adventure, 1977.
- [8] I. Cyan Worlds. Myst. <https://es.wikipedia.org/wiki/Myst>, 1993.
- [9] A. Doroudian, S. Hausknecht, and D. Kaufman. Creating an online escape room game for older adults: Needs assessment, design process, and usability testing. 2018.
- [10] E. Elumir. 13 rules for escape room puzzle design. <https://thecodex.ca/13-rules-for-escape-room-puzzle-design/>, 2018.
- [11] F. J. Gallego, R. Molina, and F. Llorens. Gamificar una propuesta docente. [https://rua.ua.es/dspace/bitstream/10045/39195/1/Gamificacio%CC%81n%20\(definicion%CC%81n\).pdf](https://rua.ua.es/dspace/bitstream/10045/39195/1/Gamificacio%CC%81n%20(definicion%CC%81n).pdf), 2014.
- [12] M. J. García, M. Hellín-Martínez, and M. Alfonso-Asencio. Experiencia educativa de escape room con personas mayores con alzheimer. https://blogcrea.imserso.es/documents/1565578/1956973/fa6be1_80fb4a50d0d84d97911017620db567e9.pdf/d472ce8e-e5d7-377e-83ef-a2a6fc9feb4c?t=1667548456872, 2020.
- [13] O. K. Heikkinen and J. Shumeyko. Designing an escape room with the experience pyramid model, 2016.
- [14] https://www.estimulacioncognitiva.info/._quÉ es la terapia de reminiscencia y cÓmo se puede realizar? https://www.estimulacioncognitiva.info/2018/04/11/qu%C3%A9-es-la-terapia-de-reminiscencia-y-c%C3%B3mo-se-puede-realizar/, 2024.

- [15] Infocom. Zork 1: The great underground empire. <https://es.wikipedia.org/wiki/Zork>, 1980.
- [16] Infocom. The hitchhiker's guide to the galaxy. [https://en.wikipedia.org/wiki/The_Hitchhiker%27s_Guide_to_the_Galaxy_\(video_game\)](https://en.wikipedia.org/wiki/The_Hitchhiker%27s_Guide_to_the_Galaxy_(video_game)), 1984.
- [17] S. A. D. J. A reminiscence program intervention to improve the quality of life of long-term care residents with alzheimer's disease: a randomized controlled trial. *Revista brasileira de psiquiatria*, 2012.
- [18] O. Labs. Job simulator. https://www.oculus.com/deeplink/?action=view&path=app/3235570703151406&ref=oculus_desktop, 2019.
- [19] F. P. Maragall. ¿qué es y qué podemos hacer para aumentar la reserva cognitiva? <https://blog.fpmaragall.org/reserva-cognitiva>, 2024.
- [20] N. Moran. Tools of immersion. <https://upthegame.nl/watch-talks/>, 2017.
- [21] D. Mulero Pérez. Interacción con objetos en realidad virtual utilizando tracking de manos, 2021-06-22.
- [22] E. e. a. Nichols. Estimation of the global prevalence of dementia in 2019 and forecasted prevalence in 2050: an analysis for the global burden of disease study 2019. *The Lancet Public Health*, 7(2), 2022.
- [23] S. Oprea, P. Martinez-Gonzalez, A. Garcia-Garcia, J. A. Castro-Vargas, S. Orts-Escalano, and J. Garcia-Rodriguez. A visually realistic grasping system for object manipulation and interaction in virtual reality environments. *Computers & Graphics*, 83:77–86, 2019.
- [24] U. D. Orueta, C. Buiza-Bueno, and J. Yanguas-Lezaun. Reserva cognitiva: evidencias, limitaciones y líneas de investigación futura. *Revista Española de Geriatría y Gerontología*, 2010.
- [25] G. Parra Caballero. Sistema de agarre realista para entornos virtuales orientado al entretenimiento, 2023-06-27.
- [26] R. C. Petersen, O. Lopez, M. J. Armstrong, T. S. D. Getchius, M. Ganguli, D. Gloss, G. S. Gronseth, D. Marson, T. Pringsheim, G. S. Day, M. Sager, J. Stevens, and A. Rae-Grant. Practice guideline update summary: Mild cognitive impairment: Report of the guideline development, dissemination, and implementation subcommittee of the american academy of neurology. *NCBI*, 2018.
- [27] M. Scrolls. Advent, colossal cave o adventure. https://it.wikipedia.org/wiki/The_Pawn, 1985.
- [28] Z. Software. Behind closed doors. <https://worldofspectrum.org/archive/software/text-adventures/behind-closed-doors-zenobi-software>, 1988.

- [29] C. Studio. Mindset. https://www.meta.com/es-es/experiences/mindset/5981642121902237/?srltid=AfmB0opZzZbDS6u0A_Ydw2YqnxQZ19XVe6VK5EA1IVpL7b0SkZwKvCg, 2023.
- [30] P. Studio. Escape simulator. <https://pinestudio.com/games/escape-simulator/>, 2021.
- [31] R. Sugar. The world is chaos. escape rooms make sense. <https://www.vox.com/the-goods/2019/8/7/20749177/escape-room-game>, 2019.
- [32] O. Systems. Mystery house. https://es.wikipedia.org/wiki/Mystery_House, 1980.
- [33] T. Takagi. Crimson room. https://archive.org/details/crimsonroom_flash, 2004.
- [34] TESOFT. Planet mephius. <https://www.generation-msx.nl/software/tesoft/legends-of-star-arthur---planet-mephius/692/>, 1985.
- [35] I. P. Tussyadiah, D. Wang, T. H. Jung, and M. tom Dieck. Virtual reality, presence, and attitude change: Empirical evidence from tourism. *Tourism Management*, 66:140–154, 2018.
- [36] M. Wiemker, E. Elumir, and A. Clare. Escape room games: Can you transform an unpleasant situation into a pleasant one? <https://dev.epicgames.com/documentation/es-mx/unreal-engine/instanciated-materials-in-unreal-engine>, 2015.
- [37] B. Woods, L. O'Philbin, E. Farrell, A. Spector, and M. Orrell. Reminiscence therapy for dementia. *Cochrane Database of Systematic Reviews*, (3), 2018.