

Escape de la mansión



Universitat
Oberta
de Catalunya

Sergio Villalba Jaén

TFG Videojuegos

Nombre Tutor/a de TF

Fernando Ruiz Velarde

**Profesor/a responsable de la
asignatura**

Paolo Gambardella

Fecha Entrega: 7 de Diciembre



Esta obra está sujeta a una licencia de
Reconocimiento-NoComercial-
SinObraDerivada [3.0 España de Creative
Commons](#)

FICHA DEL TRABAJO FINAL

Título del trabajo:	Escapa de la Mansión: Prototipo de videojuego de puzzles en Unreal Engine 5
Nombre del autor:	Sergio Villalba Jaén
Nombre del director/a:	Fernando Ruiz Velarde
Nombre del PRA:	Paolo Gambardella
Fecha de entrega (mm/aaaa):	12/2025
Titulación o programa:	Grado de Ingeniería Informática
Área del Trabajo Final:	Videojuegos
Idioma del trabajo:	Castellano
Palabras clave	Videojuegos, Unreal Engine 5, puzzles

Resumen del Trabajo

El presente Trabajo de Fin de Grado tiene como finalidad diseñar y desarrollar un prototipo de videojuego basado en la resolución de puzzles. El contexto del proyecto es la creación de una experiencia interactiva que combinan ambientación, exploración y lógica.

Este proyecto sitúa al jugador en una mansión misteriosa a la que se debe acceder tres recorriendo un camino inicial. En el interior se encontrarán diferentes habitaciones que presentan puzzles en su interior. Cada uno de los puzzles completados dará una pieza que se debe de incorporar en la puerta de salida. Cuando se hayan reunido las tres piezas el jugador escapa de la mansión y completa el juego.

La metodología de Trabajo tiene un enfoque de prototipado ágil. El desarrollo se llevará a cabo en Unreal Engine 5, motor que ha sido seleccionado gracias a su capacidad de generar entornos 3D.

Repositorio Github: <https://github.com/Sokoforfun/EscapeMansion-TFG>

Video breve del funcionamiento del juego (1 min 52 segundos):

https://www.youtube.com/watch?v=NrO9u3Yz_I4

Video extendido del funcionamiento del juego (3 minutos 18 segundos):

<https://www.youtube.com/watch?v=GLrtgp-bsCc>

Abstract

A maximum of 250 words, detailing the purpose, context of application, methodology, results and conclusions of the work

Agradecimientos

Si se considera oportuno, mencionar a las personas, empresas o instituciones que hayan contribuido en la realización de este proyecto.

Declaración de uso de recursos de terceros
Relación de Assets de Terceros

Entorno y Construcción:

- **Cave Gate Stylized** – Autor: Alzarac Store (Fab). Licencia: CC BY 4.0.
- **Mossy Cobblestone Path** – Autor: Quixel Megascans (Fab). Licencia: Standard License.
- **Sand Rock (Material)** – Autor: G4A Gaming Labs (Fab). Licencia: Standard License.
- **Rough Brick Ground Material** – Autor: Meshsmith Foundry (Fab). Licencia: Standard License.
- **Wooden Floor (Material)** – Autor: Quixel Megascans (Fab). Licencia: Standard License.
- **Damaged Brick Wall Plaster (Material)** – Autor: Quixel Megascans (Fab). Licencia: Standard License.
- **Old Tree** – Autor: Fearless Astro Star (Fab). Licencia: CC BY 4.0.
- **Simple Water Puddles (Free)** – Autor: JVAD3D Models (Fab). Licencia: CC BY 4.0.

Mobiliario y Decoración:

- **Game Ready Rustic Table With Chairs** – Autor: Hephaestus Studio (Fab). Licencia: CC BY 4.0.
- **Simple Dinner Table and Chairs** – Autor: robespierre (Fab). Licencia: Standard License.
- **Alden Bar & Counter Stools - Steel** – Autor: Toss (Fab). Licencia: CC BY 4.0.
- **Old Wooden Chair** – Autor: anthony magdelaine (Fab). Licencia: CC BY 4.0.
- **Europe Style Wooden Clock** – Autor: Chung the Artist (Fab). Licencia: CC BY 4.0.
- **Large_Antler_Chandelier_FBX** – Autor: Get Dead Entertainment (Fab). Licencia: CC BY 4.0.
- **Chandelier** – Autor: Mikhail KadiInikov (Fab). Licencia: CC BY 4.0.
- **Fireplace** – Autor: Quixel Megascans (Fab). Licencia: Standard License.
- **CCO - Doormate** – Autor: plaggy (Fab). Licencia: CC BY 4.0.
- **Billiard Pool Table** – Autor: Sakigake Furuzawa (Fab). Licencia: Standard License.

Objetos Interactivos y Props:

- **Toy Train** – Autor: Max3d (Fab). Licencia: Standard License.
- **Autonomous Robot Sweeper** – Autor: Zeps3D (Fab). Licencia: CC BY 4.0.
- **Toy FlashLight** – Autor: Sygma (Fab). Licencia: Standard License.
- **Old Rusty Key** – Autor: Alex Krush (Fab). Licencia: CC BY 4.0.
- **Valve Test** – Autor: MiljanBojovic (Fab). Licencia: Standard License.
- **CCO - Keypad Door Lock** – Autor: plaggy (Fab). Licencia: CC BY 4.0.
- **Turtle Tears Vending Machine** – Autor: EFX (Fab). Licencia: Standard License.
- **Claw Machine LOWPOLY RIGGED** – Autor: EFX (Fab). Licencia: Standard License.
- **Rusty Japanese Arcade** – Autor: Toni Garcia Vilche (Fab). Licencia: CC BY 4.0.
- **TV LOW POLY** – Autor: Oobe-Xr Srl (Fab). Licencia: Standard License.
- **Gumball Machine** – Autor: andi475 (Fab). Licencia: Standard License.
- **Water Dispenser** – Autor: Alterego (Fab). Licencia: CC BY 4.0.

Elementos Varios y Ambientación:

- **Wooden Log** – Autor: Share Textures (Fab). Licencia: CC BY 4.0.

- **Stylish Fire VFX (Free asset)** – Autor: VFXSTOCK (Fab). Licencia: Standard License.
- **Rudi Carrell Bust** – Autor: Cygnos (Fab). Licencia: CC BY 4.0.
- **Spooky Halloween Pumpkin (Scan)** – Autor: JordanFry3D (Fab). Licencia: CC BY 4.0.
- **Teapot & Cups On Table** – Autor: Polycraftstudios3D (Fab). Licencia: CC BY 4.0.
- **Human Skeleton Download FREE** – Autor: RogerSIQ3dstore (Fab). Licencia: CC BY 4.0.
- **Cruet** – Autor: dimension Dazzle (Fab). Licencia: CC BY 4.0.
- **Witch Cauldron** – Autor: Halfoun (Fab). Licencia: CC BY 4.0.
- **Poison bottle** – Autor: Pedram Ashoori (Fab). Licencia: CC BY 4.0.
- **Water Bottle Free** – Autor: elixonline (Fab). Licencia: CC BY 4.0.
- **Water Bottle** – Autor: Nicholas 3D (Fab). Licencia: CC BY 4.0.
- **Old Gravestone** – Autor: Quixel Megascans (Fab). Licencia: Standard License.
- **Stylized coffin** – Autor: Aartee (Fab). Licencia: CC BY 4.0.

Índice

Contenido

Escape de la mansión	1
Sergio Villalba Jaén	1
TFG Videojuegos	1
Nombre Tutor/a de TF	1
Fernando Ruiz Velarde	1
Profesor/a responsable de la asignatura	1
Paolo Gambardella	1
Fecha Entrega: 7 de Diciembre	1
 Esta obra está sujeta a una licencia de Reconocimiento-NoComercial- SinObraDerivada 3.0 España de Creative Commons	2
Título del trabajo:.....	3
Escapa de la Mansión: Prototipo de videojuego de puzzles en Unreal Engine 5	3
Nombre del autor:.....	3
Sergio Villalba Jaén	3
Nombre del director/a:.....	3
Fernando Ruiz Velarde	3
Nombre del PRA:.....	3
Paolo Gambardella	3
Fecha de entrega (mm/aaaa):.....	3
12/2025.....	3
Titulación o programa:.....	3
Grado de Ingeniería Informática	3
Área del Trabajo Final:.....	3
Videojuegos	3
Idioma del trabajo:.....	3
Castellano	3
Palabras clave	3
Videojuegos, Unreal Engine 5, puzzles	3
Resumen del Trabajo	3
El presente Trabajo de Fin de Grado tiene como finalidad diseñar y desarrollar un prototipo de videojuego basado en la resolución de puzzles. El contexto del proyecto es la creación de una experiencia interactiva que combinan ambientación, exploración y lógica.....	3
Este proyecto sitúa al jugador en una mansión misteriosa a la que se debe acceder tres recorrer un camino inicial. En el interior se encontrarán diferentes habitaciones que presentan puzzles en su interior. Cada uno de los puzzles completados dará una pieza que se debe de incorporar en la puerta de salida. Cuando se hayan reunido las tres piezas el jugador escapa de la mansión y completa el juego.....	3
La metodología de Trabajo tiene un enfoque de prototipado ágil. El desarrollo se llevará a cabo en Unreal Engine 5, motor que ha sido seleccionado gracias a su capacidad de generar entornos 3D.....	3
Repositorio Github: https://github.com/Sokoforfun/EscapeMansion-TFG	3
Video breve del funcionamiento del juego (1 min 52 segundos):	3
https://www.youtube.com/watch?v=NrO9u3Yz_I4	3
Video extendido del funcionamiento del juego (3 minutos 18 segundos):	3
https://www.youtube.com/watch?v=GLrtgp-bsCc	3
Abstract	3
A maximum of 250 words, detailing the purpose, context of application, methodology, results and conclusions of the work	4

Índice.....	7
Lista de figuras	11
1. Introducción	1
1.2 Contexto y justificación del Trabajo	1
1.3 Objetivos del Trabajo.....	2
Objetivo principal: Diseñar y desarrollar un prototipo jugable con puzzles en Unreal Engine 5	2
Objetivos más específicos:.....	2
1.4 Impacto en sostenibilidad, ético-social y de diversidad	2
1.5 Enfoque y método seguido	2
1.6 Planificación del Trabajo.....	2
1.7 Breve sumario de productos obtenidos.....	3
1.8 Breve descripción de los otros capítulos de la memoria	3
2. Estado del arte	3
2.1 Introducción y Definición del Ámbito	4
2.2 Análisis de referentes en videojuegos de puzzles:	4
2.2.1 The Room (Fireproof Games):	4
2.2.2 Silent Hill / Resident Evil	4
2.3 Tendencias actuales.....	4
2.4 Estudio de mercado	5
2.5 Herramientas similares:	6
2.6 Conclusiones	6
3. Materiales y métodos	7
3.1 Metodología de desarrollo:	7
3.2 Herramientas y Recursos:	7
3.3 Diseño y arquitectura	8
3.4 Gestión y Adaptación de Recursos Gráficos.....	8
4. Diseño.....	9
4.1 Análisis de requisitos y estudio de usuarios:.....	9
4.1.1 Metodología de investigación de usuarios.....	9
4.1.2 Resultados obtenidos:.....	9
4.1.3 Requisitos funcionales del sistema.	10
4.2 Arquitectura del software.	10
4.2.1 Diagrama de clases propuesto.....	10
4.2.2 Patrones de Diseño:.....	11
4.3 Diseño de mecánicas específicas:.....	11
4.3.1 Sistema de navegación del Robot.....	11
4.3.2 Diseño de físicas para los objetos pequeños	11
4.4 Diseño de niveles:	12
4.4.1 Exterior y recibidor (Zonas de transición)	12
4.4.2 Sala principal	13
4.4.3 El restaurante.....	13
4.4.4 Laboratorio.....	13
4.4.5 Discoteca	13
4.4.6 Interconexiones.....	13
5. Resultados	15

El enfoque de este apartado puede variar en función del tipo de proyecto (más de desarrollo, teórico, o de investigación). En cualquier caso, es necesario describir el resultado final alcanzado a través de la realización del trabajo final. Estos resultados deben estar alineados con las herramientas y métodos y el diseño descritos anteriormente. 15

Según el tipo de trabajo, se puede incluir una demostración o prueba de concepto, o resultados de pruebas (e.g. benchmarks, tests de usuarios) que demuestren su solidez.

15

6. Conclusiones y Trabajo futuro..... 16

6.1. Conclusiones 16

Este apartado debe incluir: 16

Una descripción de las conclusiones del trabajo: 16

Una vez obtenidos los resultados, ¿qué conclusiones se extrae? 16

Estos resultados son los esperados? ¿O han sido sorprendentes? ¿Por qué? .. 16

Una reflexión crítica sobre la consecución de los objetivos planteados inicialmente: .. 16

¿Hemos alcanzado todos los objetivos? Si la respuesta es negativa, ¿por qué? 16

6.2. Seguimiento de la planificación..... 16

Un análisis crítico del seguimiento de la planificación y metodología a lo largo del producto:..... 16

¿Se ha seguido la planificación? 16

¿La metodología prevista ha sido suficientemente adecuada? 16

¿Ha sido necesario introducir cambios para garantizar el éxito del trabajo? ¿Por qué? 16

De los impactos previstos en 1.3, ético-sociales, de sostenibilidad y de diversidad, evalúe/mencione si se han mitigado (si eran negativos) o si se han conseguido (si eran positivos). 16

Si han aparecido impactos no previstos en 1.3, evaluar/mencionar cómo se han mitigado (si eran negativos) o qué han aportado (si eran positivos). 16

6.3. Líneas de futuro 16

Las líneas de trabajo futuro que no han podido explorarse en este trabajo y han quedado pendientes. 16

7. Glosario 17

Definición de los términos y acrónimos más relevantes utilizados en la Memoria. 17

8. Bibliografía 18

Bibliografía: 18

1. Fireproof Games. The Room. Fireproof Studios; 2012. Disponible en: <https://www.fireproofgames.com/> 18

2. Konami. Silent Hill. Konami Computer Entertainment Tokyo; 1999 18

3. Statista. Revenue of the global game market from 2015 to 2023 <https://www.statista.com/> 18

4. Valve Corporation. Steam Game Stats and Market Reports <https://store.steampowered.com/> 18

5. Epic Games. Unreal Engine 5 Documentation. <https://docs.unrealengine.com/> .. 18

6. Unity Technologies. Unity Manual and Documentation <https://docs.unity.com/> ... 18

7. Godot Engine. Godot Engine Documentation. Godot Foundation; 2024. <https://docs.godotengine.org/> 18

9. Anexos 18

Anexo I: Detalle de Adaptación de Actores y Assets..... 18

Asset Original 18

Fuente 18

Modificación que se ha realizado 18

Tren de Juguete..... 18

Fab 18

Lógica y Geometría: Se ha separado la malla única en 3 componentes independientes, implementadas físicas simuladas por Spline para tener articulación en curvas.....	18
Panel con combinación.....	18
Fab 18	
Interactividad: Se ha convertido de Static Mesh a Blueprint Actor. Programación del Widget interactivo 3D con lógica de contraseña y feedback cuando se acierta.....	18
Mobiliario	18
Fab 18	
Ambientación: Retexturizado para la unificación visual y ajuste de colisiones.	18
Válvulas	18
Fab 18	
Mecánica de Blueprints y C++: Se ha implementado la lógica de rotación incremental que está vinculada a variables de estado del puzzle.	18
Iluminación	18
Varias	18
Ajustado el mapa de luz y configuración de Lumen para tener un buen rendimiento en tiempo real.....	18

Lista de figuras

Figura 1:

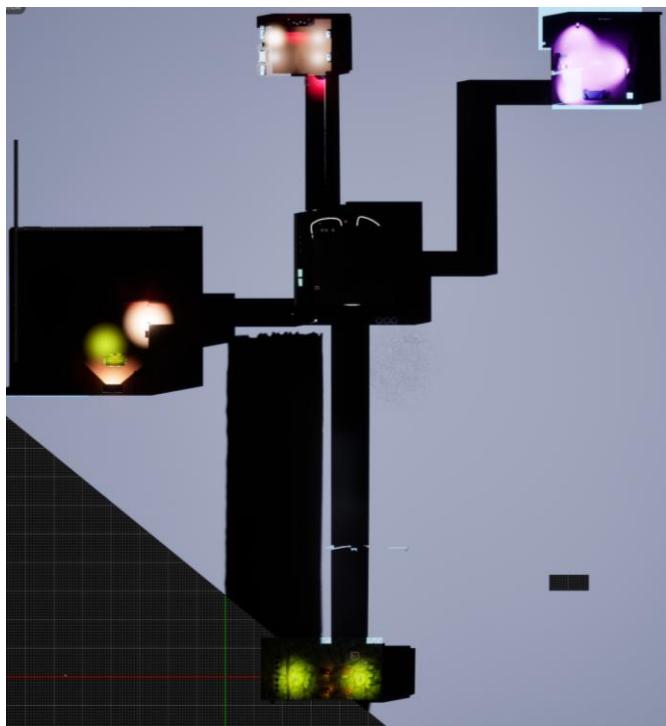


Figura 2:

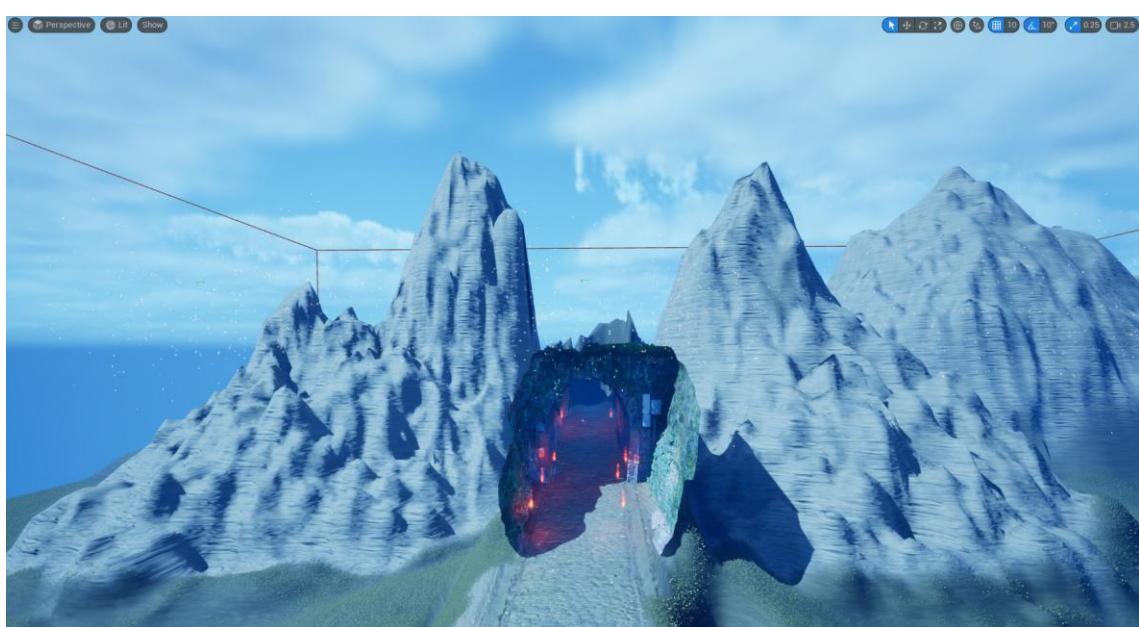


Figura 3:

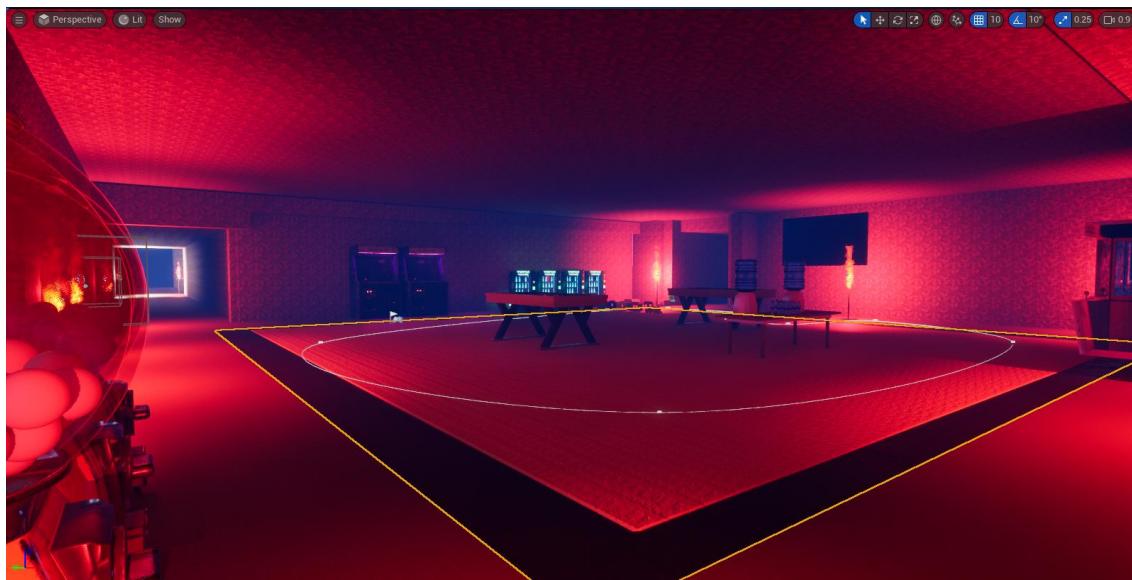


Figura 4:

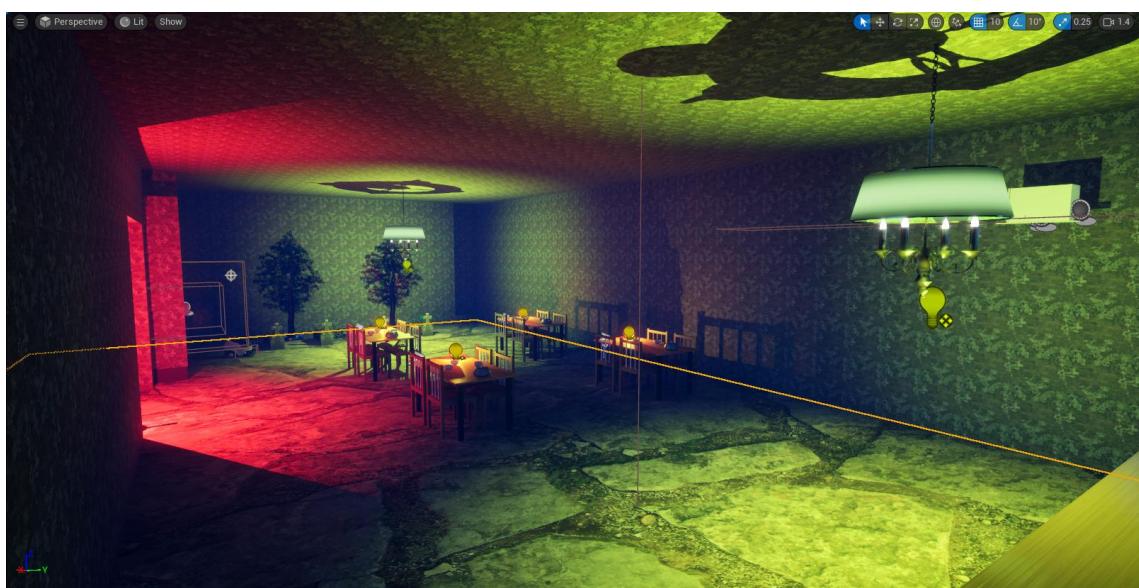


Figura 5:



Figura 6:



1. Introducción

1.1 Resumen del proyecto:

El propósito de este trabajo de fin de grado consiste en el diseño y en el desarrollo de un prototipo de videojuego que está basado en la resolución de puzzles. La propuesta tratará de situar al jugador en una mansión muy misteriosa. En el interior de la mansión el jugador deberá encontrar diferentes habitaciones que contarán con desafíos lógicos y mecánicos que deberán de ser resueltos para poder progresar en el juego.

Cada puzzle completado dará una pieza que deberá de ser puesta en la puerta de la salida. Una vez reunidas las tres piezas se desbloqueará el acceso al exterior y el juego será completado.

El desarrollo de este juego será realizado en Unreal Engine 5, este motor ha sido escogido debido a su potencia técnica y a su idoneidad para poder crear entornos 3D que sean completamente interactivos.

El proyecto implementa una arquitectura híbrida utilizando tanto C++ como Blueprints en Unreal Engine 5. Se ha diseñado un sistema escalable de interacción mediante interfaces, así como un sistema de inventario que está basado en estructuras de datos. La lógica de los diferentes puzzles utiliza patrones de diseño, así como la herencia (para los diferentes objetos interactuables) y Event Dispatchers para la comunicación entre la UI y el juego.

Repositorio del Proyecto:

<https://github.com/Sokoforfun/EscapeMansion-TFG>

1.2 Contexto y justificación del Trabajo

El desarrollo de videojuegos constituye un gran ámbito tecnológico que está en una evolución constante. Se caracteriza por mezclar arte, narrativa de forma interactiva y un gran diseño digital. Los juegos que están basados en puzzles han ido adquiriendo un papel bastante importante utilizando herramientas que fomentan el razonamiento lógico entre otras cosas.

El trabajo se centrará sobre todo en el diseño y desarrollo de un prototipo jugable en el que el jugador debe de explorar una mansión misteriosa para resolver diferentes desafíos lógicos y obtener las piezas que sean necesarias para poder terminar el juego. El proyecto es inspirado en juegos como Silent Hill donde la ambientación juegan un papel muy importante y desafía al jugador.

La elección de Unreal Engine 5 no es simplemente arbitraria. Se justifica por la necesidad de utilizar las diferentes tecnologías de la industria, el sistema de iluminación global Lumen para tener una ambientación totalmente inmersiva que es requerida para un juego de misterio además del sistema de Blueprints combinado con C++ para realizar una lógica compleja.

1.3 Objetivos del Trabajo

Objetivo principal: Diseñar y desarrollar un prototipo jugable con puzzles en Unreal Engine 5

Objetivos más específicos:

- Implementar tres puzzles independientes con un sistema de progresión (en este caso, las piezas obtenidas al resolver los puzzles)
- Diseño de la estructura de la mansión
- Realizar un vídeo de demostración que sea capaz de mostrar adecuadamente la experiencia de juego.

1.4 Impacto en sostenibilidad, ético-social y de diversidad

- Este prototipo de juego es capaz de contribuir a fomentar correctamente el pensamiento lógico y la capacidad de resolución de los diferentes problemas gracias a la interacción con puzzles. Si lo planteamos desde un punto de vista ético, este prototipo no contará con contenidos explícitos y será accesible para diversos públicos, además, será adecuado para futuras ampliaciones que puedan contemplar opciones de accesibilidad.

1.5 Enfoque y método seguido

El proyecto será abordado desde un enfoque totalmente iterativo. La metodología que se aplicará contará con ciclos de diseño, implementación y prueba. Esto permitirá implementar de forma progresiva la jugabilidad. Se ha elegido Unreal Engine 5 debido a su potencial para poder crear entornos 3D, cuenta con un sistema de Blueprints de gran potencia además de ser un motor bastante avanzado que garantiza la viabilidad técnica del proyecto.

1.6 Planificación del Trabajo

1. Planificación inicial (Semana 1): Se hará la obtención de recursos para el proyecto
2. Prototipo jugable (Semanas 2 a 6): Se hará la construcción de la mansión y desarrollo de un puzzle que sea funcional
3. Desarrollo completo del proyecto (Semanas 7 a 9): Serán incorporadas tres habitaciones con puzzles y el sistema de progresión.
4. Pruebas de jugabilidad (Semanas 10 y 11): Se realizará una evaluación interna y se harán los últimos retoques
5. Documentación (Semana 12): Elaboración de la memoria

1.7 Breve sumario de productos obtenidos

- Prototipo jugable del videojuego hecho en Unreal Engine 5
- Video de demostración de la experiencia de juego
- Memoria completa que pueda documentar el desarrollo y los diferentes resultados

1.8 Breve descripción de los otros capítulos de la memoria

- Capítulo 1: Introducción del trabajo.
- Capítulo 2: Estado del arte.
- Capítulo 3: Materiales y métodos empleados en el desarrollo del prototipo.
- Capítulo 4: Diseño del juego.
- Capítulo 5: Resultados obtenidos tras la implementación del prototipo.
- Capítulo 6: Conclusiones del trabajo.

2. Estado del arte

2.1 Introducción y Definición del Ámbito

Los videojuegos de puzzles y exploración representan un género muy consolidado que se basan en la interacción constante del jugador con el entorno para poder resolver los enigmas y avanzar en el juego. El presente TFG entra en la categoría de Escape Rooms Digitales donde la clave son los ambientes inmersivos y la integración de las diferentes mecánicas. El desafío técnico de este proyecto es implementar una arquitectura de software modular que nos permita la creación rápida de nuevos puzzles sin tener que reescribir la lógica base.

2.2 Análisis de referentes en videojuegos de puzzles:

2.2.1 The Room (Fireproof Games):

Este juego no se analiza solo por su estética sino por su interacción táctil. El referente para este TFG es cómo el jugador manipula directamente los elementos del escenario (introducir llaves, deslizar paneles...). Este proyecto adapta esto a una vista en primera persona utilizando diferentes mecánicas como el trazado de rayos para poder simular la precisión del ratón sobre los objetos 3D del escenario.

Aportación al TFG: Nuestro proyecto es capaz de emular esa sensación utilizando la interfaz de Blueprint (BP_Interaction) que es combinada con Raycasting (LineTraceByChannel). Este sistema garantiza que una única pulsación active comportamientos únicos y precisos en cada uno de los elementos del escenario (llaves, paneles de la combinación, válvulas...).

2.2.2 Silent Hill / Resident Evil

Estos juegos utilizan la gestión de recursos (llaves, ítems limitados...) como un sistema de bloqueo de progreso del jugador. La clave no es solo almacenar los recursos sino usar el inventario como un mecanismo de paso para que el jugador tenga que cumplir ciertos objetivos antes de continuar en la historia.

Aportación al TFG:

La arquitectura del inventario está basada en estructuras de datos en lugar de una colección de objetos. Cada uno de los objetos tiene metadatos (Nombre del objeto, icono y clase del objeto) que son capaces de permitirle al sistema implementado verificar la existencia de cierto objeto para desbloquear puzzles específicos. (Por ejemplo, hay un robot que tan solo se activa si existe la batería de este en el inventario).

2.3 Tendencias actuales

La elección de Unreal Engine 5 está totalmente justificada por la capacidad de optimizar el pipeline de desarrollo siguiendo las diferentes tendencias del mercado indie:

- Iluminación dinámica (Lumen): Se ha aprovechado el sistema de Lumen (iluminación global) que es capaz de modificar la iluminación en tiempo real sin requerir al proceso de baking (precálculo de luces). Esto nos permite una iteración de diseño de nivel rápida y cambios dinámicos en la atmósfera del juego, esto es un elemento muy clave en los juegos de terror y misterio.
- Desarrollo híbrido (Blueprint / C++): Esta es la tendencia actual en el desarrollo de videojuegos.
 - o Blueprints: Se utiliza para la lógica de alto nivel (flujo de puzzles, UI, Event Dispatchers) por su rapidez de prototipado.
 - o C++: Es utilizado para la lógica compleja y optimizada (por ejemplo, el cálculo de la rotación de las válvulas), esto garantiza el rendimiento donde es más crítico.
- Control de inputs: Se adopta el nuevo sistema de inputs de Unreal Engine 5 que nos permite la creación de contextos de mapeo de inputs, esto nos facilita el remapeo de las teclas y la gestión de entradas específicas tanto para la UI como para la interacción dentro del juego. Además, también se usa el sistema de inputs antiguo para ciertas acciones.

2.4 Estudio de mercado

El análisis de mercado actual para videojuegos de puzzles en primera persona nos indica una viabilidad comercial bastante positiva debido a los siguientes puntos:

- Coste de Producción: Al centrar el desarrollo en el diseño de niveles y en la lógica de programación apoyándose en algunos assets de bibliotecas existentes (como Quixel Megascans) se reducen los costes y tiempos de producción artística.
- Público objetivo: Existe una base de usuarios bastante amplia y fiel al género de misterio y terror que es capaz de valorar adecuadamente la calidad de los desafíos lógicos.
- Rentabilidad: Plataformas de distribución digital como Steam muestran una alta tasa de éxito en los diferentes juegos indie de puzzles que estén bien ejecutados, a su vez, este tipo de juegos tienen un gran potencial para el streaming.

2.5 Herramientas similares:

- Unity: Es bastante popular debido a su curva de aprendizaje y al uso de C#, el sistema de scripting visual no está tan integrado como los Blueprints de Unreal y su pipeline de renderizado requiere una mayor configuración manual que Unreal.
- Godot: Se trata de una buena alternativa open source y eficiente pero actualmente carece de tecnologías nativas que sean equivalentes a Lumen o Nanite para poder alcanzar un nivel de realismo visual que esté requerido por la dirección artística del proyecto sin que haya un esfuerzo desproporcionado.

Unreal Engine 5 ha sido seleccionada finalmente por ofrecer una solución robusta para tener un gran renderizado de alta fidelidad, físicas y un buen sistema de gameplay que permite centrar adecuadamente el esfuerzo del TFG en la programación de las diferentes mecánicas implementadas.

2.6 Conclusiones

El análisis que se ha realizado confirma la viabilidad técnica y comercial del proyecto. Se ha identificado correctamente que la combinación de mecánicas de puzzles clásicas con tecnologías actuales (Lumen de UE5) tiene un nicho de mercado muy rentable y atractivo.

La comparativa frente a las otras herramientas dice que la elección de Unreal Engine 5 como la solución óptima es una buena decisión debido a su arquitectura híbrida (Blueprints y C++) y a sus sistemas nativos de física e iluminación debido a que reducen significativamente los tiempos de desarrollo en comparación con Unity o Godot permitiendo así centrar el esfuerzo del proyecto en la complejidad lógica y en la arquitectura del software

3. Materiales y métodos

3.1 Metodología de desarrollo:

Para la ejecución del proyecto se ha adoptado de forma correcta una metodología de desarrollo iterativa e incremental. Como se trata de un prototipo que tiene un plazo temporal acotado se ha evitado el modelo de cascada para favorecer ciclos cortos de desarrollo que nos permitan validar de forma correcta las mecánicas de forma aislada e integrarlas en el sistema principal.

El ciclo de vida del desarrollo ha sido dividido en tres fases principales:

1. Fase de mecánicas claves: Se ha implementado la arquitectura base del jugador (movimiento, sistema de interacción mediante Raycasting) y clases padres como por ejemplo ObjetoInteractuable.
2. Fase de sistemas: Aquí se han desarrollado los subsistemas modulares (un inventario que está basado en estructuras de datos y sistema de puzzles con herencia).
3. Contenido y pulido: Diseño de nivel, implementación de puzzles específico como el Robot y las válvulas además de corrección de bugs que han ido surgiendo como con las físicas y las colisiones.

3.2 Herramientas y Recursos:

El entorno de trabajo se ha configurado de la siguiente manera:

- Motor de Juego: Unreal Engine 5.4.4
- Entorno de desarrollo integrado: Visual Studio 2022
- Control de versiones: Git, que está alojado en Github. Se ha configurado un fichero llamado .gitignore que es específico para Unreal Engine, se han excluido las carpetas de archivos intermedios y binarios como binaries, DerivedDataCache, Intermediate y Saved para poder optimizar adecuadamente el peso del repositorio.
- Software de apoyo: OBS Studio para la captura de las pruebas de jugabilidad y documentación del proyecto de manera visual.

3.3 Diseño y arquitectura

El proyecto se ha estructurado siguiendo una arquitectura orientada a objetos para garantizar la escalabilidad:

- Estructura de directorios: Se sigue el estándar de la nomenclatura de Unreal Engine (Prefijos “BP_” para blueprints, “WBP_” para Widgets Blueprints, “M_” para materiales...) organizando los diferentes assets por tipo y funcionalidad.
- Modularidad: Los diferentes puzzles se desarrollan como actores independientes que son capaces de comunicarse con el jugador y el mundo mediante el uso de Interfaces y Event Dispatchers evitando así dependencias fuertes.

3.4 Gestión y Adaptación de Recursos Gráficos

Para la ambientación visual del proyecto, se ha optado por una estrategia de Asset Flipping de manera optimizada utilizando recursos de librerías externas (Fab, Quixel Megascans) para el diseño de niveles. Sin embargo, estos recursos no se han implementado de forma directa, sino que han sido sometidos a un proceso de adaptación técnica para poder alinearlos correctamente con las mecánicas y la ambientación del juego.

A continuación, se detallan algunas de las intervenciones que se han realizado sobre los assets principales:

- Tren (Puzle de colores): El asset original llamado “Toy Train” se trataba de una locomotora que incluía únicamente una malla estática con una locomotora y un vagón fusionados. Para poder añadirle movimiento realista en las curvas se ha utilizado la herramienta de modelado de Unreal Engine para separar la geometría en componentes individuales. Seguidamente, se ha creado una jerarquía de actores en Blueprint y se ha programado un sistema de movimiento basado en Splines matemáticos que permite a cada vagón rotar de forma independiente simulando así físicas de arrastre. Además, se han creado instancias de material dinámicas para asignar un color diferente a cada vagón lo cual es necesario para la resolución del puzzle.
- Panel numérico: Se ha partido a través de un modelo 3D estático de un panel numérico. Se le ha programado una interfaz de usuario totalmente original que gestiona la lógica de contraseñas, da feedback visual y sonoro además de un sistema de apertura de puertas.
- Ambientación e iluminación: Los diferentes elementos decorativos se han unificado visualmente mediante un sistema de iluminación dinámica (Lumen) que cuentan con tonalidades específicas que aportan coherencia estética.
- Foco de Discoteca (BP_JuegoLuces): Se ha creado un actor compuesto a partir de una SpotLight y una cápsula de colisión. Para poder darle movimiento se ha optado por un componente RotatingMovement desplazando el eje de pivote logrando así que el foco orbite por la sala.

4. Diseño

4.1 Análisis de requisitos y estudio de usuarios:

Para definir las especificaciones del prototipo se ha simulado una fase de toma de requisitos que es basada en el diseño centrado en el usuario. Se ha planteado un estudio que ha sido cualitativo y cuantitativo para poder identificar las diferentes preferencias del público objetivo del género de puzzles y terror psicológico.

4.1.1 Metodología de investigación de usuarios.

Se ha diseñado un cuestionario digital que ha sido dirigido a diferentes usuarios habituales de juegos como The Room, Resident Evil o Silent Hill. Los objetivos de este estudio fueron los siguientes:

- Identificar correctamente los puntos de interés habituales en este tipo de juego de puzzles.
- Evaluar las diferentes preferencias que hay entre los puzzles lógicos y los puzzles físicos.
- Determinar el nivel de tolerancia y la frustración en la búsqueda de objetos dentro del mundo.

4.1.2 Resultados obtenidos:

- Una buena cantidad de usuarios dice que sienten frustración cuando hay que encontrar algún objeto que está escondido o que los objetos interactuables son muy pequeños o difíciles de seleccionar con el ratón.
 - Medida tomada: Se ha implementado un sistema de Raycasting con ayuda visual y colisiones bastante simplificadas y fáciles de acertar con el ratón. Los objetos pequeños (Llaves, linternas...) tienen una colisión significativamente superior a la del tamaño del objeto para poder agarrarlos con facilidad.
- Los usuarios dicen que prefieren que los Widget que se utilicen en los paneles de la combinación sean sencillos de hacer click.
 - Medida tomada: Al pulsar la tecla "E" sobre un panel de la combinación se abrirá el widget para poner la combinación correcta.
- Feedback visual: Los usuarios dicen que les gustaría que exista una confirmación visual o auditiva cuando una acción es correcta.
 - Medida tomada: Cuando se acierta un puzzle o se averigua la combinación correcta de un panel. Saldrá un texto en el que ponga "Código Correcto" o habrá alguna pista visual que nos indique que el puzzle ha sido resuelto como por ejemplo que aparezca una llave o se desbloquee cierta cerradura o puerta.

4.1.3 Requisitos funcionales del sistema.

Basándose en el análisis anterior se han definido los siguientes requisitos funcionales:

- Interacción: El sistema debe de permitir al jugador interactuar correctamente con los diferentes objetos del entorno a distancia mediante un puntero central, se debe de poder diferenciar entre recoger objetos y operar con los mecanismos del entorno.
- Inventario: El inventario no debe de ser únicamente un contenedor, debe de poder verificar la posesión de los diferentes ítems específicos para poder habilitar interacciones en el mundo (por ejemplo, el Robot que requiere de una batería).
- Persistencia del estado: Los puzzles deben de reconocer el estado en el que están. Por ejemplo, en caso de que un puzzle haya sido resuelto, este debe permanecer resuelto y que no se reinicie.

4.2 Arquitectura del software.

El diseño técnico está orientada a la Programación Orientada a Objetos.

4.2.1 Diagrama de clases propuesto.

Se ha propuesto una arquitectura que está basada en clases padre y clases hijos para el correcto funcionamiento del proyecto.

- Clase Base (ObjetoInteractuable): Es la clase que se encarga de definir la interfaz de comunicación. Esta gestionará correctamente la respuesta básica al Raycast.
 - o Subclase Pickups: Estos objetos pertenecientes a esta clase están heredando la estructura de la clase padre implementando así la lógica para añadir el objeto a la estructura de datos del inventario y a destruir el actor en la escena cuando este se añada al inventario. En esta subclase hay objetos como BP_Linterna y BP_Bateria (Se tratan de una linterna y la batería para el robot mencionado anteriormente).
- Clase Blueprint base BP_Keypad: Se trata de un panel de la combinación programado para poder abrir puertas con una pequeña animación.
- Clase Nativa (ValveBase): Se ha diseñado una clase en C++ destinada para los puzzles de rotación, esto aprovecha correctamente la eficiencia matemática del lenguaje para los cálculos de ángulos y transmite la información al Blueprint.

- BP_JuegoLuces: Se trata de un actor autónomo que gestiona la lógica del puzzle de la discoteca. Se ha implementado una máquina de estados mediante temporizadores para alternar entre sí es seguro (color verde), sino lo es (color rojo) o si va a comenzar a no ser seguro (naranja), comprobando la posición del jugador para validar si el juego se está completando adecuadamente o no.
- BP_PuertaSalida: Se trata de un actor que gestiona si se ha cumplido el objetivo del juego o no. Se comunica con el personaje para lograr verificar el estado de la progresión del juego (número de llaves que se tienen) y activa la interfaz de fin del juego en caso de que el objeto se haya cumplido.

4.2.2 Patrones de Diseño:

- Event Dispatchers: Estos se utilizan para desacoplar la lógica de los puzzles de sus consecuencias. Los elementos interactivos (paneles, válvulas...) emiten un evento al resolverse y los elementos del nivel (puertas, spawners...) escucharán a estos eventos y harán la acción pertinente.
- Máquinas: El comportamiento del NPC (Robot) Se modelará mediante una máquina de estados simple. Detectará si está apagado o si está encendido, en caso de estar encendido sigue una ruta (mediante splines) y cuando la termina finaliza el recorrido y se vuelve a parar. El jugador puede volver a pulsar la E sobre el robot para que vuelva a realizar la ruta.

4.3 Diseño de mecánicas específicas:

4.3.1 Sistema de navegación del Robot

Para el puzzle de observación de patrones se ha decidido descartar el uso de IA navegacional debido a su imprevisibilidad. Se diseña un sistema determinista que está basado en Splines asegurando así que el robot sea capaz de pasar exactamente por los puntos de interés en un orden que sea narrativo y controlado. El movimiento se debe de interpolar utilizando Timelines para tener una mayor fluidez.

4.3.2 Diseño de físicas para los objetos pequeños

Para poder anticipar problemas de usabilidad con objetos pequeños (llaves) que simulan físicas se ha planteado un diseño de colisión híbrido. La malla visual debe de gestionar la caída física mediante Simulate Physics mientras que se forzará mediante un Construction Script la adhesión de una caja de colisión que esté sobredimensionada respecto al objeto original para así poder garantizar que el jugador sea capaz de interactuar con el objeto sin ningún tipo de problema.

4.3.3 Mecánica de Atención

Para la sala de la discoteca se ha diseñado una mecánica basada en el tiempo y el movimiento. Este desafío obliga al jugador moverse de forma constante.

- Estados: El foco alterna colores con tiempos ya definidos
- Se utiliza un sistema de contadores internos para ver si el jugador permanece en la luz roja (el contador se reinicia) o si está en la luz verde (el contador suma).
- Al alcanzar 3 aciertos el sistema (utilizando un SpawnActor) da la llave de la sala automáticamente y destruye el foco dando así a entender que el juego ha terminado.

4.3.4 Recogida automática y condición de victoria

Con el fin de agilizar el juego y evitar errores, las llaves se recogerán de forma automática sin que el jugador pueda descartarlas utilizando el sistema de inventario.

- Recogida por contacto: Las llaves utilizan un evento de superposición para sumarse automáticamente a una variable llamada "LlavesConseguidas" y se autodestruye.
- Puerta final: La puerta de salida comprueba si el número de llaves es igual o superior a 3. En caso de que la condición se cumpla, el juego se detiene y activa el widget "WBP_FinJuego".

4.4 Diseño de niveles:

El diseño espacial del proyecto ha sido realizado en un único nivel persistente que se ha estructurado pensando en que el jugador pueda tener una referencia espacial de manera constante minimizando así la desorientación en un entorno de terror y misterio mientras se fomenta la exploración de manera no lineal limitada según el progreso del jugador (sin conseguir cierta combinación no se abrirá cierta puerta por lo que el jugador no puede explorar el área que se encuentra detrás de la puerta sin haber conseguido la combinación).

A continuación, se detalla el diseño funcional de cada área:

4.4.1 Exterior y recibidor (Zonas de transición)

El nivel comienza en una zona exterior que está acotada por una montaña que actúa como barrera natural impidiendo así la huída y focalizando la atención del jugador en una única vía de acceso disponible.

- Transición: El acceso al interior de la mansión se hace mediante un descenso vertical por lo que no se puede volver a la superficie. Gracias a esto se cumple una función narrativa debido a que el jugador queda atrapado.
- Recibidor: Una vez dentro, esta sala actúa como una zona de aterrizaje y tutorial. Se trata de un espacio seguro que es lineal y que es capaz de enseñar al jugador los controles básicos de movimientos antes de entrar a la sala principal.
- El exterior se puede ver en [Figura 2]

4.4.2 Sala principal

Es el nodo central de navegación. Desde esta zona se ramifican los accesos a las otras tres salas principales.

- Narrativa: En esa sala se encuentra la puerta de salida final que actuará como recordatorio del objetivo del juego.
- La sala principal se puede ver en [Figura 3]

4.4.3 El restaurante

Diseñado para la mecánica del Robot camarero.

- Diseño: Tiene una distribución de mesas que permite ver la ruta del robot sin obstáculos.
- Iluminación: Se centra la atención en el robot que está en reposo para poder tenerlo en cuenta como punto de interés. Se ha colocado el panel numérico en la pared cercana. Cuando este se resuelve se obtiene una llave.
- El restaurante se puede ver en [Figura 4]

4.4.4 Laboratorio

Se cambia la estética a un tono más industrial y de investigación.

- Mecánica: Aquí está el puzzle de las válvulas. Se contemplan indicadores visuales que indican en tiempo real la cantidad de veces que se debe de girar cada válvula (esta lógica está implementada en C++).
- El laboratorio se puede ver en [Figura 5]

4.4.5 Discoteca

Esta zona introduce un cambio drástico en la atmósfera, se utiliza la iluminación como mecánica principal.

- Diseño del puzzle: Se debe seguir un foco durante cierto tiempo para poder completar el puzzle. El jugador debe salirse del foco cuando este se ilumina en rojo y debe entrar cuando se ilumina en verde
- La discoteca se puede ver en [Figura 6]

4.4.6 Interconexiones

Finalmente, con el fin de comprender la interconexión de las áreas que se han descrito anteriormente se presenta la distribución del nivel [Figura 1].

5. Resultados

El enfoque de este apartado puede variar en función del tipo de proyecto (más de desarrollo, teórico, o de investigación). En cualquier caso, es necesario describir el resultado final alcanzado a través de la realización del trabajo final. Estos resultados deben estar alineados con las herramientas y métodos y el diseño descritos anteriormente.

Según el tipo de trabajo, se puede incluir una demostración o prueba de concepto, o resultados de pruebas (e.g. benchmarks, tests de usuarios) que demuestren su solidez.

6. Conclusiones y Trabajo futuro

6.1. Conclusiones

Este apartado debe incluir:

Una descripción de las conclusiones del trabajo:

- Una vez obtenidos los resultados, ¿qué conclusiones se extrae?
- Estos resultados son los esperados? ¿O han sido sorprendentes? ¿Por qué?

Una reflexión crítica sobre la consecución de los objetivos planteados inicialmente:

- ¿Hemos alcanzado todos los objetivos? Si la respuesta es negativa, ¿por qué?

6.2. Seguimiento de la planificación

Un análisis crítico del seguimiento de la planificación y metodología a lo largo del producto:

- ¿Se ha seguido la planificación?
- ¿La metodología prevista ha sido suficientemente adecuada?
- ¿Ha sido necesario introducir cambios para garantizar el éxito del trabajo? ¿Por qué?

De los impactos previstos en 1.3, ético-sociales, de sostenibilidad y de diversidad, evalúe/mencione si se han mitigado (si eran negativos) o si se han conseguido (si eran positivos).

Si han aparecido impactos no previstos en 1.3, evaluar/mencionar cómo se han mitigado (si eran negativos) o qué han aportado (si eran positivos).

6.3. Líneas de futuro

Las líneas de trabajo futuro que no han podido explorarse en este trabajo y han quedado pendientes.

7. Glosario

Definición de los términos y acrónimos más relevantes utilizados en la Memoria.

8. Bibliografía

Bibliografía:

1. Fireproof Games. The Room. Fireproof Studios; 2012. Disponible en:
<https://www.fireproofgames.com/>
2. Konami. Silent Hill. Konami Computer Entertainment Tokyo; 1999
3. Statista. Revenue of the global game market from 2015 to 2023
<https://www.statista.com/>
4. Valve Corporation. Steam Game Stats and Market Reports
<https://store.steampowered.com/>
5. Epic Games. Unreal Engine 5 Documentation.
<https://docs.unrealengine.com/>
6. Unity Technologies. Unity Manual and Documentation
<https://docs.unity.com/>
7. Godot Engine. Godot Engine Documentation. Godot Foundation; 2024.
<https://docs.godotengine.org/>

9. Anexos

Anexo I: Detalle de Adaptación de Actores y Assets

Asset Original	Fuente	Modificación que se ha realizado
Tren de Juguete	Fab	Lógica y Geometría: Se ha separado la malla única en 3 componentes independientes, implementadas físicas simuladas por Spline para tener articulación en curvas.
Panel con combinación	Fab	Interactividad: Se ha convertido de Static Mesh a Blueprint Actor. Programación del Widget interactivo 3D con lógica de contraseña y feedback cuando se acierta.
Mobiliario	Fab	Ambientación: Retexturizado para la unificación visual y ajuste de colisiones.
Válvulas	Fab	Mecánica de Blueprints y C++: Se ha implementado la lógica de rotación incremental que está vinculada a variables de estado del puzzle.
Iluminación	Varias	Ajustado el mapa de luz y configuración de Lumen para tener un buen rendimiento en tiempo real.

