UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

GRADO EN DISEÑO DE VIDEOJUEGOS

PROYECTO FIN DE GRADO

TUTORIZACIÓN IMPLÍCITA EN VIDEOJUEGOS

Kai Xian Lan Ji

Dirigido por

Luis Gutiérrez Tamurejo

CURSO 2023-2024

TÍTULO: TUTORIZACIÓN IMPLÍCITA EN VIDEOJUEGOS

AUTOR: Kai Xian Lan Ji

TITULACIÓN: GRADO EN DISEÑO DE VIDEOJUEGOS

DIRECTOR/ES DEL PROYECTO: Luis Gutiérrez Tamurejo

FECHA: junio de 2024

RESUMEN

Este trabajo de fin de grado explora las diferentes técnicas y maneras de tutorizar de forma implícita un videojuego. El principal objetivo de este trabajo es conseguir implementar estas técnicas en un videojuego de puzzles. Para ello, se ha llevado a cabo una investigación sobre el género de puzzles y los distintos tipos de tutoriales que existen; y un análisis exhaustivo de tres títulos exitosos de videojuegos de puzzle: *The Witness, Portal* y *The Talos Principle*.

Los resultados de esta investigación revelaron que hay que tener en cuenta el esquema de controles del videojuego, la curva de dificultad y aprendizaje, asociar elementos visuales con mecánicas y utilizar objetos de escena para guiar al jugador. Todas estas técnicas se implementaron en el proyecto que se estaba desarrollando en paralelo.

Finalmente, se realizó una fase de pruebas con compañeros, amigos y profesores para comprobar la efectividad de estas técnicas. Los resultados fueron positivos, obteniendo que el 80% consiguieron completar el videojuego.

Palabras clave: *videojuegos, videojuegos de puzzles, tutoriales de videojuegos, diseño de videojuegos, aprendizaje, técnicas de tutorización implícita*

ABSTRACT

This final degree project explores different techniques and ways to implicitly tutor a videogame. The main objective of this project is to develop these techniques in a puzzle game. To achieve this, research was conducted on the puzzle genre and the different types of tutorials that exist; as well as an exhaustive analysis of three successful puzzle game titles: *The Witness, Portal,* and *The Talos Principle.*

The results of this research revealed that it is important to consider the game's control scheme, the difficulty and learning curve, associate visual elements with mechanics, and use scene objects to guide the player. All these techniques were implemented in the project that was being developed in parallel.

Finally, a testing phase was carried out with colleagues, friends, and professors to verify the effectiveness of these techniques. The results were positive, with 80% managing to complete the game.

Keywords: *videogames, puzzle videogames, videogames tutorials, game design, learning, implicit tutoring in videogames*

TABLA RESUMEN

	DATOS
Nombre y apellidos:	Kai Xian Lan Ji
Título del proyecto:	Tutorización implícita en videojuegos
Directores del proyecto:	Luis Gutiérrez Tamurejo
El proyecto se ha realizado en colaboración de una empresa o a petición de una empresa:	NO
El proyecto ha implementado un producto: (esta entrada se puede marcar junto a la siguiente)	SI
El proyecto ha consistido en el desarrollo de una investigación o innovación: (esta entrada se puede marcar junto a la anterior)	NO
Objetivo general del proyecto:	Desarrollar un videojuego de puzzles con técnicas de tutorización implícita

Índice

Índice de Figuras

Índice de Gráficas

Palabras clave

Asset – Representación de cualquier objeto que se pueda ser utilizado en un juego

Testeo – Probar el videojuego

Loop – Ciclo, en este caso, jugable del videojuego

Gameplay – Se refiere a la jugabilidad del videojuego

Shader - Sombreador

SFX – Efectos de sonido

Sprint – Periodo de tiempo en el que se trabaja para completar una cantidad de tareas en una metodología Scrum

Capítulo 1. INTRODUCCIÓN

Este trabajo está centrado en la tutorización implícita del jugador a lo largo de un videojuego de puzzles. Por ello, se ha realizado una investigación exhaustiva sobre este tema analizando diferentes títulos del género de puzzles. Los resultados y conclusiones obtenidas se utilizarán en el desarrollo de un videojuego del mismo género. En él, el jugador aprenderá a jugar por sí solo con únicamente elementos implícitos y diegéticos que se hayan implementado.

1.1 Planteamiento y justificación del problema

¿Es posible guiar al jugador únicamente empleando el entorno y el diseño de niveles?

En los últimos años, los videojuegos han tenido una tendencia más casual. Como consecuencia, los desarrolladores han tenido que implementar ciertas técnicas para guiar al jugador paso a paso. Sin embargo, muchas de estas técnicas son a veces demasiado explícitas (pop-ups, o ventanas emergentes, de textos o vídeos explicando mecánicas nuevas, pistas de audio directas o indirectas explicando qué hacer en cada momento, etc..). Esto muchas de las veces rompen totalmente la inmersión de juego e incluso puede llegar a ser molesto hacia el propio jugador.

Por tanto, cada vez se publican menos videojuegos que intenten guiar al jugador de manera inmersiva, aunque siguen existiendo. Entonces, si se recogen las técnicas que emplean estos videojuegos, ¿es posible guiar al jugador únicamente empleando el entorno y el diseño de niveles?

1.2 Objetivos del proyecto

1.2.1 Objetivo general

Con el objetivo de responder a la pregunta de la hipótesis, se desarrollará una demo jugable de un videojuego de puzzles lo más cercano a un producto final. Esta demo incluirá una forma de guía al jugador sin ninguna forma de pista explícita empleando el propio diseño de niveles y el entorno. Además, estará respaldada por la investigación previa de diferentes técnicas y herramientas que emplean diferentes juegos del mismo género para guiar al jugador.

1.2.2 Objetivos específicos

Con el fin de cumplir con el objetivo general del proyecto, se requerirá de unos objetivos más específicos. Estos objetivos están divididos en dos grandes grupos: el videojuego y la investigación.

Los objetivos del videojuego son todas las features (o características de juego) que se quieren implementar. Las más globales son: el controlador, los puzzles, el arte/escenarios y la UI.

Por otro lado, los objetivos de la investigación son el análisis de diferentes títulos de videojuegos de puzzles y la psicología del jugador.

Asimismo, se quiere plantear, como objetivo específico, cumplimentar uno de los Objetivos de Desarrollo Sostenible de las Naciones Unidas, u ODS. En este caso, se pretende implementar ajustes y configuraciones de accesibilidad para personas con daltonismo. Con ello, se intenta cumplimentar y apoyar el "Objetivo 10: Reducción de las Desigualdades", concretamente el punto 10.2 que dice lo siguiente: "*De aquí a 2030, potenciar y promover la inclusión social, económica y política de todas las personas, independientemente de su edad, sexo, discapacidad, raza, etnia, origen, religión o situación económica u otra condición"* (Naciones Unidas, 2015).

Todos estos objetivos están desglosados en algunos más particulares. A continuación, se mostrará el desglose general de los objetivos específicos:

Videojuego:

- Controlador
  - Mecánica principal. Recoger y disparar colores a elementos de escena.
  - Movimiento básico primera persona.
- Puzzles
  - Diseño de puzzles.
  - Diseño de niveles.
  - Diseño de elementos adicionales.
  - Implementar técnicas de tutorización, guía y pistas.
- Arte/Escenarios
  - Shader Unlit.
  - Modelos.
- Audio
  - SFX de juego
  - SFX de UI
  - Música
- UI
  - Menú principal
  - Menú de pausa
  - HUD
- Ajustes
  - Ajustes de audio
  - Ajustes gráficos
  - Ajustes de juego
  - Ajustes de accesibilidad

Investigación:

- Análisis e investigación de diferentes títulos del género de puzzles.
  - Analizar progresión de dificultad de puzzles.
  - Estudio de métodos de pistas diegéticas empleadas para el apoyo hacia el jugador.

    - Análisis de la narrativa.
- Análisis de videojuegos con ajustes de accesibilidad.
- Psicología del jugador
    - Psicología de color

Capítulo 2. ANTECEDENTES

2.1 Contexto

En las últimas décadas ha surgido una nueva forma de entretenimiento audiovisual diferente a lo conocido en aquella época como películas, obras de teatro, libros etc... A esta nueva forma se le conoce como videojuegos. A diferencia de los otros medios de entretenimiento mencionados anteriormente, los videojuegos introducen un elemento crucial, la interactividad de usuario con el medio.

La entrada de los videojuegos generó un gran impacto en la industria del entretenimiento, provocando que millones de personas, tanto consumidores como inversores, se interesen en el producto. Como consecuencia, se provocó un gran boom, o auge repentino, que conllevó a una rápida evolución de los videojuegos. Con ello, se comenzaron a establecer los primeros géneros de videojuegos.

2.1.1 Géneros de videojuegos

Una de las revistas más famosas de la época "*Computer Gaming World*", en 1981, usaba tres categorías para referirse a los géneros: arcade, wargames (o juegos de guerra), y aventura. Esto con los años se expandió a una lista de géneros más diversa, transformando la lista en: estrategia, aventura, RPG (Role-playing game, o juegos de rol) de aventura, wargame, y acción/arcade.

Como se puede ver, a medida que pasa el tiempo, se crean más y más géneros, pero no existe ningún convenio que estipule, de manera generalizada, cuáles son de verdad. Los géneros de videojuegos son una ilusión, para una persona un género puede ser un subgénero de otra, un simple sabor o un aliño para la ensalada (Dominic Arsenault, 2009).

Además, si ya es complicado establecer los géneros, clasificar los propios videojuegos en estos géneros lo es más aún. Es muy común que un título beba mucho de otros. Esto supone que se generen nuevos géneros de videojuegos como el caso de los "metroid-vania", que es un género que combina mecánicas (lo que puedes hacer dentro de un videojuego) de juegos de la saga *Metroid* y de la saga *Castlevania;* o los "souls", que es un género que se creó con la popularidad y aparición de los juegos de la saga "Dark Souls" de FromSoftware.

Asimismo, aunque un videojuego contenga elementos de cierto género no tiene por qué pertenecer a éste. Esto ocurre muy a menudo con los puzzles. En prácticamente todos los títulos grandes hay algún matiz de puzzles, por ejemplo: *The Legend of Zelda: Tears of Kingdom* (Nintendo EPD, Monolith Soft, 2023), con sus santuarios; la saga *Uncharted* (Naughty Dog, 2007), con sus puzzles para abrirte paso a siguientes salas; la saga *God of War* (Santa Monica Studio, 2005), con sus puzzles para alcanzar zonas; y muchos más títulos.

Esto simboliza que los puzzles son algo de sumo valor a la hora de diseñar y desarrollar videojuegos, pero entonces ¿en qué se diferencia un videojuego de puzzles con los puzzles que puede haber en videojuegos de otros géneros?

2.1.2 El género de puzzles

Los puzzles han estado presente entre nosotros desde hace muchísimos años, y hoy día tienen muchas finalidades. Pueden llegar tanto a mejorar capacidades cognitivas como pensamiento crítico o visión espacial, como incluso servir de entretenimiento para las personas (Hanna R. Rodenbaugh, Heidi L. Lujan et al., 2014). Esto último también se les aplica a los videojuegos de puzzles.

Respondiendo a la pregunta, la diferencia principal es que en un videojuego de puzzles, el conflicto principal no es entre jugador y los personajes, sino que averiguar una solución, la cual suele estar ligada a resolver enigmas, aprender cómo utilizar diferentes herramientas, y la manipulación o reconfiguración de objetos (Mark Wolf, 2001, p. 113-134).

A continuación, en la Figura 1, se puede apreciar un esquema sencillo que refleja las diferencias entre los videojuegos de puzzles y otros videojuegos con algunas secciones con puzzles. En él se puede ver que los videojuegos de puzzles son aquellos donde la solución de problemas es el núcleo y mecánica principal, y donde la principal fuente de satisfacción del jugador es resolver estos problemas (Marçal Mora-Cantallops, 2018).

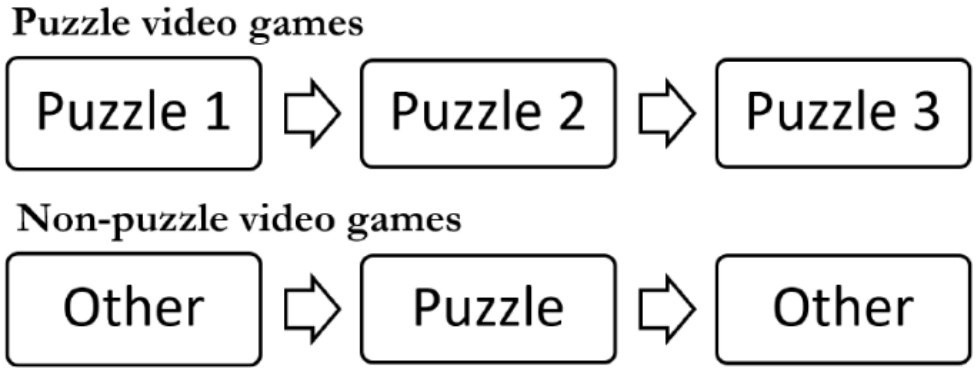


Figura 1. Los videojuegos de puzzles avanzan de puzzle en puzzle mientras que los videojuegos que no son de puzzles tienen otras acciones críticas o actividades (Marçal Mora-Cantallops, 2018). (Fuente: extraído de https://www.researchgate.net/figure/Puzzle-video-games-transition-from-puzzle-to-puzzle-while-non-puzzle-video-games-have_fig1_327639714)

Por otro lado, también existen varios factores o características que predominan en videojuegos de este género. Esto no supone que todo videojuego de puzzles ha de cumplir estos "requisitos", pero sí suelen ser muy comunes en la gran mayoría de títulos. Estas características o features son principalmente las siguientes:

- Interruptores y cables: Los interruptores son elementos interactuables que activan mecanismos. Los cables, en cambio, son elementos visuales en el entorno que conectan los interruptores con los mecanismos. Normalmente, estos mecanismos son puertas que llevan al jugador al siguiente nivel. Un claro ejemplo es *Portal* (Valve Corporation, 2007), los interruptores son botones rojos y grandes que están conectados a puertas mediante un "cable" formado por puntos (Fig. 2).

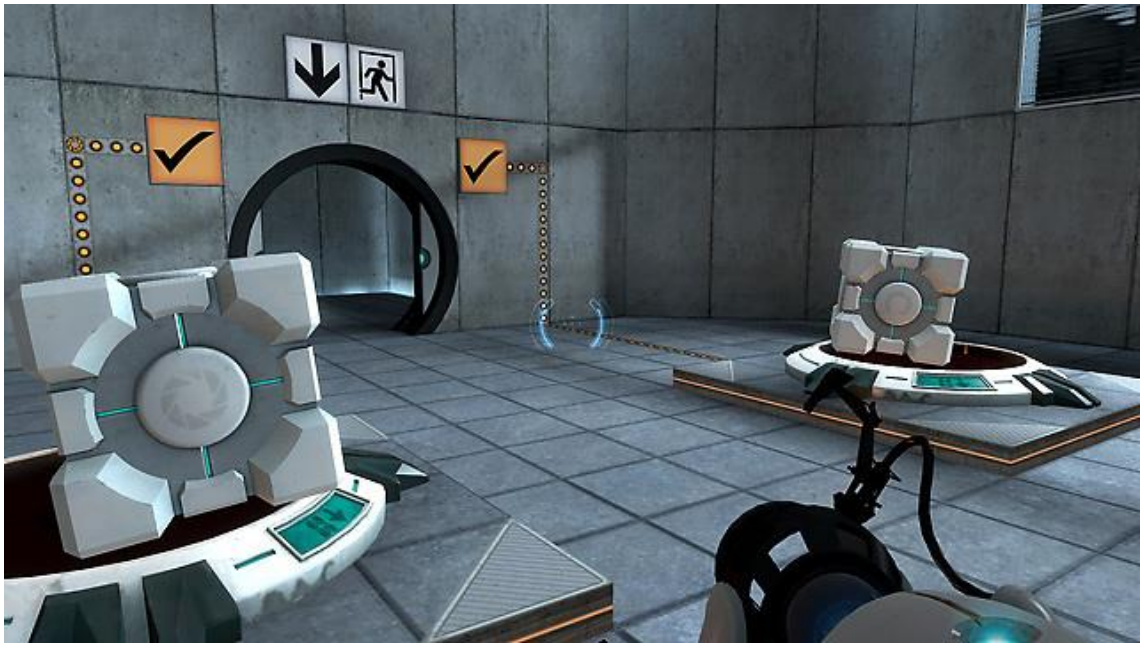


Figura 2. Ejemplo de interruptor y cable en Portal (Valve Corporation, 2007). (Fuente: extraído del videojuego Portal)

- Puertas. Las puertas son una herramienta importante que sirve para bloquear el acceso al jugador a ciertos lugares o, directamente, el avance al siguiente nivel.

- Recoger y mover objetos. Esta mecánica o habilidad, es muy común en los videojuegos de puzzle. Sin irnos muy lejos, en el ejemplo de la Figura 2 de *Portal,* se puede ver un caso de uso de esta característica. En este ejemplo, los objetos que se pueden mover y recoger son los cubos. Con la fuerza de la gravedad, al moverlos encima de los interruptores, sirven como peso para abrir la puerta.

- "Easter eggs" o secretos ocultos. Los easter eggs, son elementos ocultos implementados en medios como cómics, películas, vídeos e incluso en otras ramas como en el Microsoft Excel 97. Sin embargo, están normalmente más asociados a los videojuegos (Mago, 2019).

- Acertijos o adivinanzas. Debido a la similitud de "juego" entre los puzzles y los acertijos o adivinanzas, es usual encontrarlos como pista para completar puzzles. En la Figura 3, se puede ver un ejemplo en *The Talos Principle* (Croteam, 2015) de acertijo para recoger un coleccionable

Figura 3. Ejemplo de acertijo en The Talos Principle. (Fuente: extraído del videojuego The Talos Principle)

Finalmente, aunque cada género tiene sus características principales, existen puntos en común que son imprescindibles a la hora de enseñar y guiar a los jugadores: el game design (o diseño de juego) y los tutoriales.

2.1.3 Game design y tutoriales

El game design es el acto de decidir cómo debería de ser un juego, en otras palabras, se trata del conjunto de decisiones (las reglas, el pacing o ritmo de juego, el concepto riesgo-recompensa, etc..) que se tienen que tomar a la hora de desarrollar un juego (Jesse Schell, 2008). Además, esto también conlleva a pensar en qué puede y no puede hacer el jugador, es decir, las mecánicas de juego. Estas mecánicas son estas reglas o métodos invocados por agentes al interactuar con el mundo de juego (Sicart, 2008).

Asimismo, este concepto tiene una alta conexión con los tutoriales. Los tutoriales son una parte esencial de un videojuego. Son estrategias y herramientas empleadas por los diseñadores de videojuegos para mejorar la jugabilidad de los propios videojuegos (Shuangyuan C., Fang L., 2022). Estos tutoriales son importantes por el mero hecho de que normalmente es la primera toma de contacto del jugador con el producto, sirviendo como portada y enganche a lo que va a experimentar.

Es por ello por lo que se debe recalcar la importancia que tienen estos conceptos en un videojuego. Actualmente, según explica Matthew M. White en su libro *Learn to Play: Designing Tutorial for Videogames*, existen tres tipos de tutoriales y/o guía: los opcionales, las *Flashcards* o tutoriales explícitos y los que no son tutoriales como tal o tutoriales implícitos.

Los tutoriales opcionales, como su nombre indica, no tienen por qué completarse. Es simplemente como un "modo de juego" extra que se puede saltar, aunque siempre es recomendable (Matthew M. White, 2014).

Las *flashcards* o tutoriales explícitos son aquellos en los que los diseñadores han optado por utilizar elementos como textos, pistas de audio, imágenes, vídeos, y manuales de instrucciones. Al utilizar estos recursos, los jugadores son menos propensos a perderse o malentender mecánicas, ya que reciben esta información quieran o no, ya que son las más "ofensivas". Lo ideal sería no abusar de estos elementos más allá de mostrar los esquemas de controles en pantalla o explicar una mecánica muy compleja que sea difícil de explicar (Matthew M. White, 2014).

Los "no tutoriales" o tutoriales implícitos, son formas de enseñar al jugador que evitan utilizar estos elementos explícitos. La idea principal es simular un aprendizaje natural y orgánica sin apenas ninguna explicación. Sin embargo, si no se consigue un buen resultado es probable que el jugador se pierda y se encuentre bloqueado por lo que el jugador se podría cansar del juego y no jugarlo más, sobre todo en jugadores novatos que recién han comenzado a jugar a videojuegos. Además, muchas veces los desarrolladores dan por sentado que el jugador ya ha jugado a: videojuegos como tal, y que alguno de estos videojuegos hayan sido del mismo género que el videojuego en cuestión. Poniendo un ejemplo de tutoriales implícitos, en el título *Bayonetta* del estudio *PlatinumGames*, al jugador nada más empezar, se le suelta en medio de un escenario donde, sin saber nada del videojuego, ha de derrotar a todos los enemigos que se aproximan.

Debido a la tendencia más casual en los videojuegos, los tutoriales tienden a ser más explícitos que implícitos. Los diseñadores quieren que el jugador no se pierda en ningún concepto sacrificando parte de la experiencia de aprendizaje de juego con el objetivo de llegar a más público.

Un claro ejemplo de esto son los "souls" de la desarrolladora *FromSoftware*. Este estudio japonés es muy famoso por sus videojuegos con una dificultad elevada. Su saga más representativa, *Dark Souls* (FromSoftware, 2011), apenas tiene tutoriales. Simplemente utilizan algunos textos al inicio para enseñar al jugador los controles y poco más. A continuación, en la Figura 4, se puede ver un ejemplo:

Figura 4. Ejemplo de texto tutorial de Dark Souls III. (Fuente: extraído del videojuego Dark Souls III)

Sin embargo, en *Elden Ring* (FromSoftware, 2022), su último título galardonado con el premio al mejor juego del año 2022, a parte de estos pocos textos para los controles, también aparecen una ventanas emergente o pop-ups (ejemplo en la Figura 5). Además de salir la ventana, también se congela el tiempo para que se reduzcan las probabilidades de que el jugador se salte el texto.

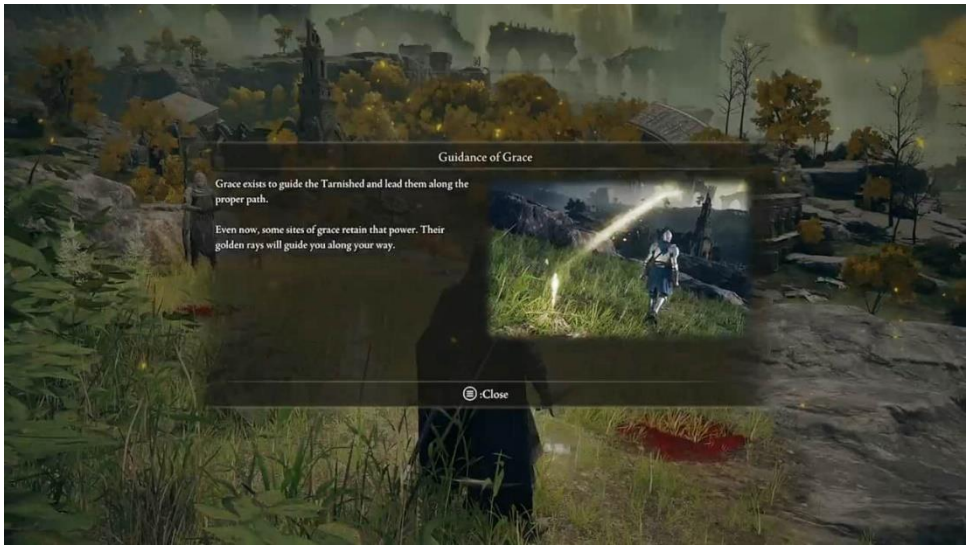


Figura 5. Ejemplo de ventana emergente tutorial en Elden Ring. (Fuente: extraído del videojuego Elden Ring)

Como se puede ver en esta comparativa, existe una evolución más casual en lo que se refiere a tutorización. No por ello se quiere decir que esté bien o esté mal, ya que como se ha explicado previamente, el diseño de juego son decisiones, y cada cual tendrá su explicación. En este caso podría ser que querían aumentar su rango de público objetivo, debido a que, según Hidetaka Miyazaki, director de *Elden Ring*, este había sido su título más ambicioso hasta la fecha.

No obstante, en el género de puzzles, se puede encontrar que la gran mayoría de desarrolladores prefieren optar por una guía sin textos ni vídeos, solamente que el entorno y diseño de niveles hable por sí solo, es decir, una guía y tutorial implícita.

Es cierto que en los últimos años (desde 2017) ha crecido el número de estudios y artículos sobre tutoriales y enseñanza en los videojuegos, ya que previo a este año no ha habido mucha investigación sobre este tema (Shuangyuan C., Fang L., 2022). Sin embargo, si se centra la búsqueda en tutoriales implícitos, los resultados tienden a ser más escasos.

2.2 Análisis de tutorización de referentes en videojuegos de puzzles

Para recopilar información sobre cómo videojuegos de puzzles en primera persona guían al jugador, se realizará un análisis y estudio profundo sobre la tutorización de tres grandes títulos de este género: *The Witness, The Talos Principle, Portal*. Se tendrá en cuenta diversos factores. Entre ellos se encuentran: los tutoriales, el loop principal jugable (o el ciclo que se repite en cada puzzle), y el pacing de aprendizaje y dificultad.

Este último apartado es de los más importantes a la hora de diseñar este tipo de videojuegos. Se puede resumir en cuatro puntos:

1. Las mecánicas principales se introducen de forma separada e independiente.
2. Estas mecánicas se introducen con puzzles básicos y sencillos.
3. El jugador tiene la oportunidad de practicar e integrar esas mecánicas con otras en puzzles más avanzados.
4. Los puzzles van aumentando la complejidad hasta que otra mecánica sea introducida.

Estos puntos han sido sacados del artículo *Learning curves: analysing pace and challenge in four successful puzzle games* (Linehan, Bellord, Kirman, H. Morford, & Roche, 2014)

2.2.1 *The Witness*

The Witness (Thekla Inc., 2016) es un videojuego de puzzles en primera persona donde el jugador se encuentra en una isla llena de misterios y puzzles a resolver. Se trata de un videojuego híbrido que cumplimenta funciones principales como exploración y puzzles, pero a su vez, obvia la narrativa y controles complejos (Bonner, 2016).

Se trata de un videojuego sin ningún tipo de guía ni tutorial explícito, es decir, el jugador debe aprender todo por sí solo. Esto abarca desde los controles como el movimiento y el giro de cámara, hasta el botón para interactuar con la mecánica base del videojuego (solo existe una mecánica core, o mecánica central). Todo esto es posible por la simpleza del controlador como tal, al igual que el juego que se pretende realizar, cuanto más simple los controles sean, más fácil de explicar y más fácil de entender.

La aventura comienza en un lugar cerrado. Solo es posible interactuar con un elemento, una puerta con un panel (Fig. 6). Para abrir la puerta es necesario interactuar con el panel, y con ello, se aprende la mecánica principal del videojuego: deslizar con el dedo líneas y formas que

funcionan como interruptor. Además, aunque no se explica esta zona funciona como tutorial implícito.

Figura 6. Primer puzzle y toma de contacto con The Witness. (Fuente: extraído del videojuego The Witness)

Con esto, *The* Witness, enseña al jugador las mecánicas más básicas de forma independiente y aislada. Como bien estipulan Linehan y sus compañeros, es un punto crucial si se quiere hacer una experiencia de aprendizaje orgánica.

Por otra parte, este título también cumple con el cuarto punto mencionado previamente. Los puzzles siempre van de lo más básico y simple para enseñar nuevas mecánicas, hasta puzzles más difíciles y complejos (Fig. 7). Estos últimos puzzles suelen combinar diferentes mecánicas.

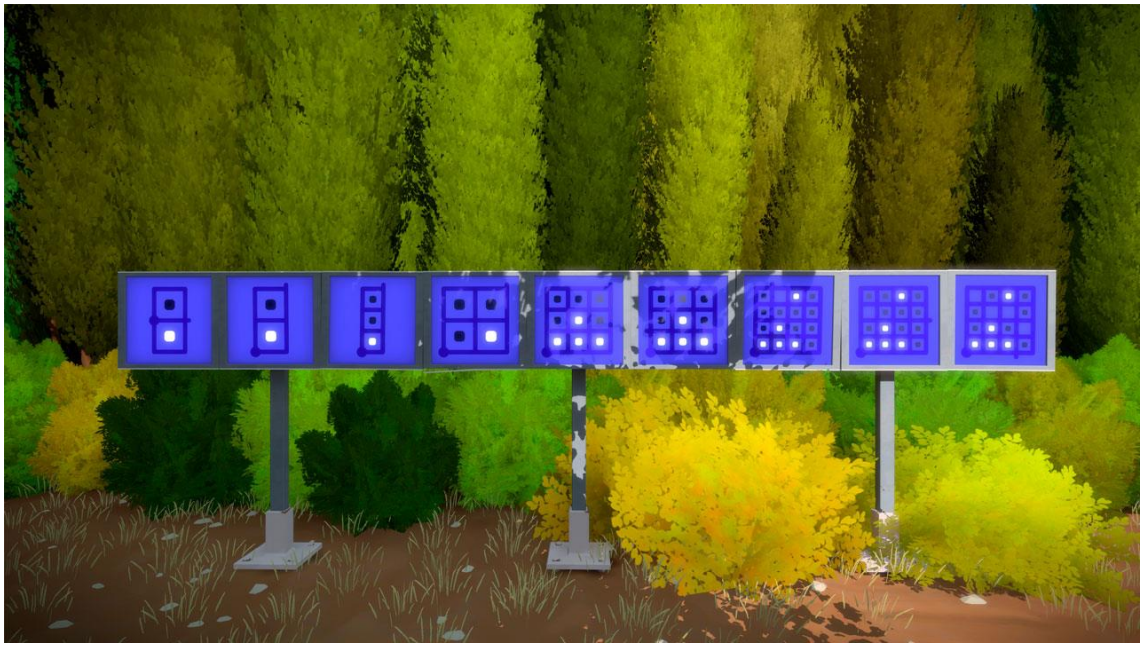


Figura 7. Ejemplo de evolución de dificultad de los puzzles en The Witness. (Fuente: extraído del videojuego The Witness)

Asimismo, *The Witness* utiliza medios visuales e incluso auditivos para ayudar al jugador a resolver puzzles. Es bastante común que en el videojuego se usen elementos de entornos para generar pistas. En el caso de la Figura 8, se utiliza una rejilla de metal para generar sombras sobre un panel. Con ello, si se sigue el camino que dicta la luz, se podrá completar el puzzle.

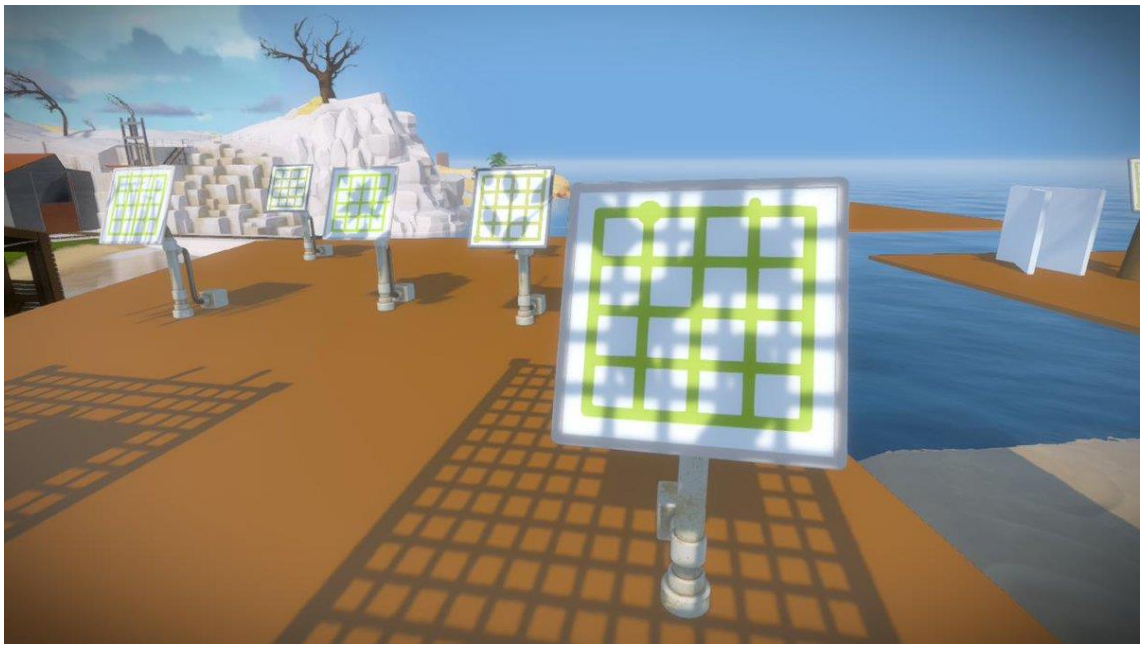


Figura 8. Ejemplo de pista visual usando las sombras. (Fuente: extraído del videojuego The Witness)

En cuanto al loop principal de videojuego, consistiría en, grosso modo, completar un panel o conjunto de paneles para abrir puertas o activar otros paneles para continuar. Con ello, el objetivo principal es completar un "monumento" que daría acceso a la sección de puzzles finales.

Siempre hay feedback visual y auditiva que hace saber al jugador que lo que está haciendo está bien o mal. Además, los paneles están conectados con cables que se pueden seguir.

2.2.2 *Portal* y *Portal 2*

Portal (Valve Corporation, 2011), es uno de los videojuegos más icónicos de toda la historia de los videojuegos de puzzles. Se trata de un videojuego que combina puzzles con portales y una narrativa llena de emociones.

Respecto a la tutorización, en *Portal,* se explican los controles con pequeños paneles con el propio esquema de control (Fig. 9). Por ello, no se trataría de un tutorial implícito, sino explícito. Aun así, dentro de los tutoriales explícitos, se encontraría dentro de un grupo muy cercano a los implícitos. Esto se debe a que los paneles explicativos no van más allá de la explicación de los controles. El resto de aprendizaje se le propone al jugador con el propio diseño de los niveles.

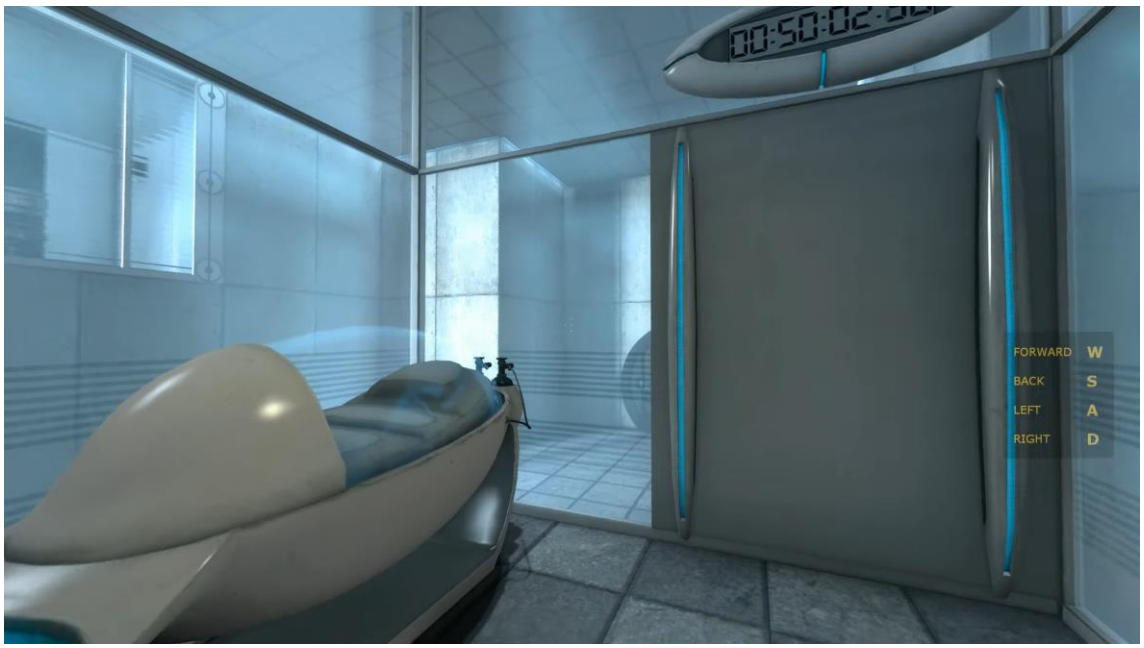
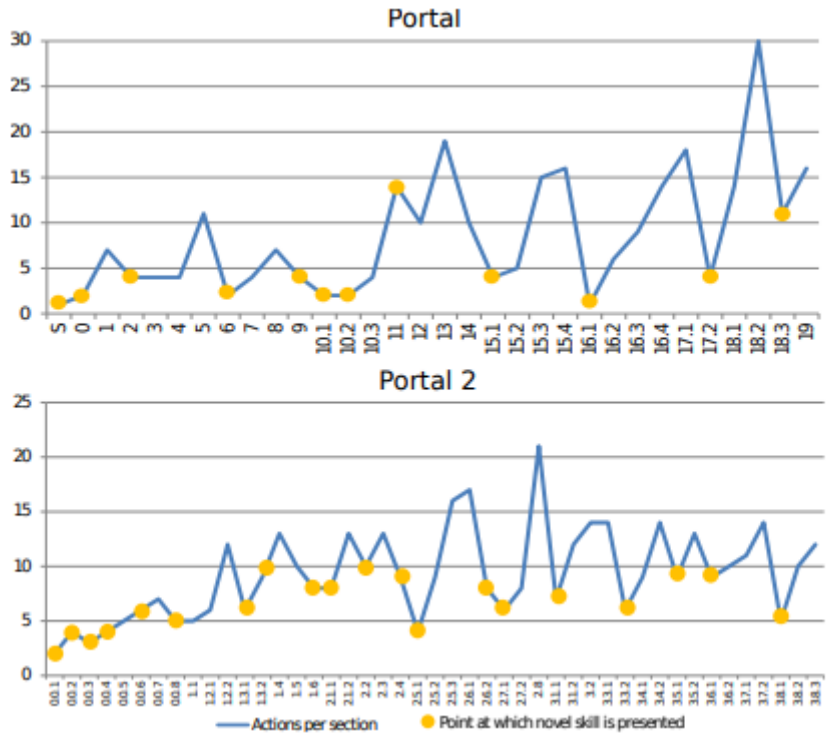


Figura 9. Esquema de control del tutorial en Portal. (Fuente: extraído del videojuego Portal)

Al igual que en *The Witness*, en *Portal*, el jugador comienza en un lugar cerrado. Además, el videojuego está divido por salas. Siguiendo la comparativa con *The Witness*, la introducción de las nuevas mecánicas está ligada a los puntos mencionados sobre la curva de aprendizaje. En la *Gráfica 1*, se pueden observar dos gráficas. Ambas gráficas muestran las acciones requeridas por

sección para completar un puzzle. Asimismo, está incluido los puntos en los que se introduce una mecánica nueva en el videojuego. Se puede observar claramente que siempre que explica una mecánica nueva, hay un decrecimiento en las acciones requeridas. Esto es así para hacer que el jugador aprenda y entienda la mecánica nueva de una forma más sencilla y fácil.



Gráfica 1. Gráfica de acciones requeridas por nivel en Portal y Portal 2. (Fuente: extraído de https://www.researchgate.net/figure/Minimum-number-of-player-actions-required-to-solve-successive-puzzles-in-Portal-top_fig1_268389769)

Cabe destacar que no sólo se emplea esta técnica de ir a lo más básico al introducir una mecánica nueva. En *Portal 2,* como menciona Eva Cid en su libro *Portal o la ciencia del videojuego*, cuando el videojuego pasa de laboratorios y salas cerradas a un entorno mucho más abierto, el jugador se puede sentir desorientado. Por ello, los primeros puzzles en esta zona nueva, los diseñadores decidieron hacer que el jugador reaprenda las mecánicas en un ambiente totalmente diferente (Cid, 2016).

Por otra parte, especialmente en *Portal 2*, se juega y utiliza mucho en contraste, luces e incluso música para guiar al jugador por el entorno o para mostrar que ha acertado (Cid, 2016). En la Figura 10, se puede ver claramente un gran contraste de blancos y negros, siendo los blancos son lugares en los que se pueden poner portales.

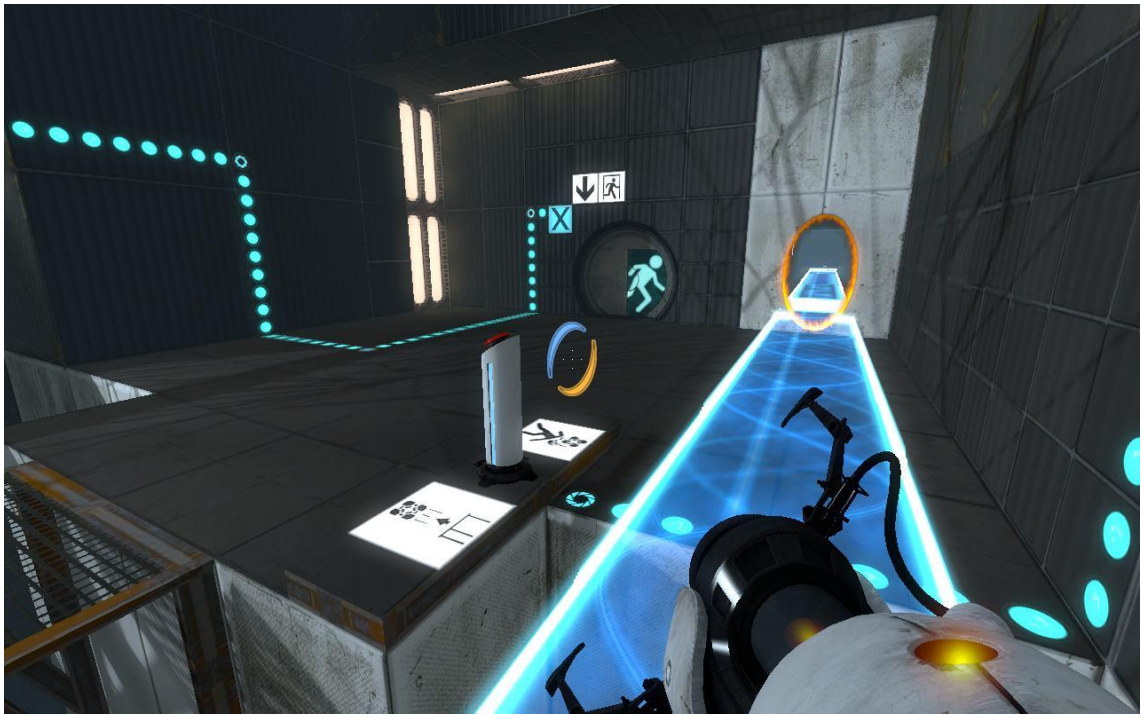


Figura 10. Ejemplo de contraste en los niveles para guiar al jugador en Portal 2. (Fuente: extraído del videojuego Portal 2)

Respecto al loop principal, en *Portal*, sería buscar la puerta, activar y abrir la puerta, y avanzar al ascensor al siguiente nivel o sala. A diferencia de *The Witness*, no hay un monumento que completar. El videojuego es lineal, las salas vienen una detrás de otra. Al igual que en *The Witness,* siempre hay feedback visual y auditiva que hace saber al jugador que lo que está haciendo está bien o mal. También existen "cables" que llevan al mecanismo que abre la puerta como bien se ha mostrado anteriormente.

2.2.3 Talos principle

Talos Principle (Croteam, 2014), es un videojuego de puzzles en primera persona que combina un gran diseño de puzzles con una narrativa llena de filosofía.

En cuanto a la tutorización, apenas se explica nada. Al igual que el *The Witness*, los controles básicos de movimiento no se explican ni se enseñan los controles. Sin embargo, a pesar de que las mecánicas específicas tampoco se explican con textos o audios, sí que es cierto que al acercarse a los elementos interactuables se puede ver con qué botón se interacciona, es decir, el esquema de control. Por ejemplo, al acercarse a un "Jammer" o bloqueador, sale un pequeño texto diciendo "Clic izquierdo" para recoger (Fig. 11).

Figura 11. Ejemplo de interfaz al interactuar con un elemento interactuable. (Fuente: extraído del videojuego The Talos Principle)

The Talos Principle, también siguen la línea de cada vez que una mecánica es implementada se explica con puzzles más sencillos y básicos. Sin embargo, en este título en concreto, se le da libertad al jugador a realizar puzzles en el orden que quiera (respetando siempre la zona o sección en la que se encuentra).

En este videojuego existen lo que se llaman “sigils”, o sellos y emblemas. Estos vienen en tres colores:

- Verde. Sirven para desbloquear zonas nuevas. Son obligatorios.
- Amarillos. Sirven para desbloquear mecánicas nuevas. Son obligatorios.
- Rojos. Sirven para desbloquear desafíos en “La Torre de desafíos de Elohim”. Son opcionales

El esquema de colores, aparte de diferenciar el uso de cada sigil, sirve para distinguir las dificultades de los puzzles. Cabe destacar también que al igual que en *The Witness*, *The Talos Principle*, se basa en “completar” un monumento con estos sigils (Fig. 12).

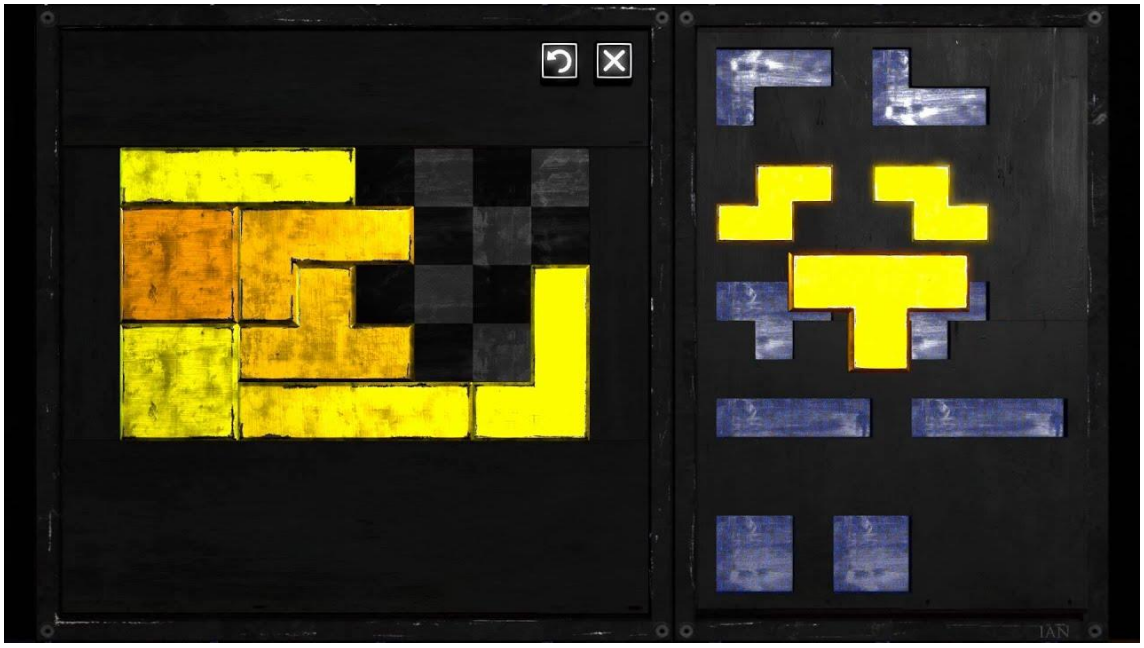


Figura 12. Ejemplo de monumento a completar con los Sigils amarillos. (Fuente: extraído del videojuego The Talos Principle)

Finalmente, en cuanto al loop principal, consistiría en completar puzzles para conseguir sigils para completar monumentos y avanzar. Además, al igual que los otros dos títulos analizados, siempre hay feedback visual y auditiva que hace saber al jugador que lo que está haciendo está bien o mal.

Capítulo 3. DESARROLLO DEL PROYECTO

3.1 Planificación del proyecto

Respecto a la planificación, se ha creado un diagrama de Gantt con las diferentes tareas a realizar para completar el desarrollo del producto. Está dividido en las diferentes etapas de un desarrollo de videojuegos: preproducción, producción y postproducción.

La primera fase, o preproducción, se realizarán principalmente tareas que requieran de pensar y planificar. También, se desarrollarán algunos elementos de prueba como el *shader unlit* para la visual o algún *blocking* (o prototipo con bloques simples) inicial para ver cómo sería un nivel.

Luego, a finales de enero y a comienzos de febrero, se plantea comenzar con el desarrollo. En esta fase se implementará todo lo necesario para tener una demo jugable de principio a fin.

Finalmente, en los últimos meses, se quiere dejar un margen para refinar y arreglar los pequeños fallos. Además, es en esta fase donde se implementarán tanto la configuración como los ajustes de accesibilidad de daltonismo mencionados en los objetivos específicos.

Es importante anotar que a medida que vaya avanzando el desarrollo del proyecto, las tareas se desglosarán en algunas más específicas, aumentando la cantidad y el tamaño del diagrama de Gantt mostrado en la Figura 13. Asimismo, cabe destacar que, durante todo el proceso de desarrollo del videojuego, existirá un testeo exhaustivo tanto por el propio autor como de personas ajenas para conseguir un feedback variado.

	octubre	noviembre	diciembre	enero	febrero	marzo	abril	mayo	junio
PREPRODUCCIÓN									
Brainstorming									
Anteproyecto									
Diseño de puzzles/niveles									
Blocking nivel prueba									
Shader Unlit									
PRODUCCIÓN									
HUD									
Sistema de escenas									
Implementación controlador									
Blocking niveles									
Implementación niveles									
POSTPRODUCCIÓN									
UI (Menús y configuración)									

Figura 13. Diagrama de Gantt del desarrollo del proyecto. (Fuente: elaboración propia)

3.2 Desarrollo del producto

En esta sección se explicará con detalle y con capturas todo el proceso que ha conllevado para el desarrollo del proyecto planteado previamente. Para ello, se seguirá una estructura similar a un desarrollo de videojuegos divididas en tres grandes bloques: Preproducción, producción y postproducción.

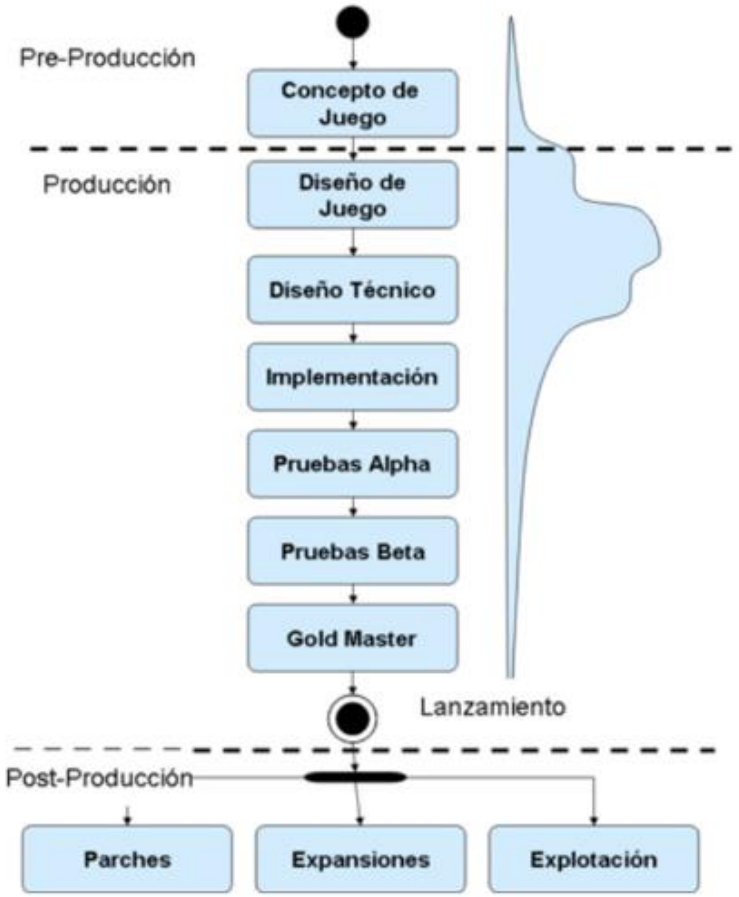


Figura 14. Esquema de fases de un desarrollo de videojuegos desglosado (Pereira, A. M. M., 2014). (Fuente: extraído de: https://pdfs.semanticscholar.org/68f0/0648ad5b3dbcda557d9eee40da10e8733a08.pdf)

Una preproducción, que engloba todo el apartado de conceptualización y preparación del proyecto. En este caso, también se realizó una búsqueda de material que se iba a emplear en el desarrollo, como vídeos o assets (elementos como modelos o shaders)

La producción del proyecto. Esta es la etapa que más tiempo conlleva. Esto se debe porque va desde el diseño de juego (en este caso de los puzzles y niveles) hasta la implementación. Además, es común realizar "sprints", es decir, dividir la producción en más etapas más pequeñas con el fin de obtener versiones del videojuego más pequeñas con las que poder hacer pruebas. Estas versiones suelen ser, como bien se indica en la Figura 14, el Alpha, la Beta, y lo que se considera la versión final, la Gold Master.

Finalmente se encuentra la postproducción. Normalmente en esta fase se realizan parches para arreglar posibles bugs y fallos críticos de lanzamiento y se comienza la parte de crear contenido extra o extensiones. Estas extensiones por sí solas se pueden considerar como un desarrollo entero con las mismas fases. (Pereira, A. M. M., 2014).

Sin embargo, debido a que este proyecto es de menor escala, no se ha planteado en ningún momento desarrollar contenido extra por lo que se utilizará esta fase como una fase para implementar aquellos elementos que no son cruciales para que el videojuego funcione correctamente. Estos elementos son principalmente elementos de interfaz de usuario y ajustes como bien se ha descrito en el diagrama de Gantt establecido en el apartado anterior.

3.3 Preproducción

3.3.1 Conceptualización

Tal y como se ha comentado, la primera fase de todo desarrollo de videojuegos es el proceso de conceptualización. Desde un principio, se tenía claro que se quería desarrollar un videojuego de puzzles. Sin embargo, la temática no estaba tan clara. Por ello, lo primero que se realizó fue un brainstorming para sacar un listado de ideas de videojuegos de puzzles. El listado fue el siguiente:

1. Videojuego de puzzles cuya mecánica principal sea conectar cables de un lado a otro. Puzzles del tipo portal con salas. Realista.
2. Videojuego de puzzles que se basa en la perspectiva, luces y sombras. “Mundo abierto”, pero no tan grande. Lowpoly.
3. Videojuego de puzzle relacionado con colores con puzzles básicos de completar este puzzle para ir a otro y completar el siguiente, como si fuese un “Escape Room”. Una sola sala.
4. Videojuego de puzzles cuya mecánica principal sea el cuentagotas de Photoshop, los colores tienen diferentes propiedades. Puzzles por salas. Lowpoly, pero más simple con shader.

Tras este listado, tocó elegir la idea a desarrollar. Pese a ser una de las que subjetivamente más gustaban, se descartó la segunda idea de perspectiva en un “mundo abierto”. Este fue debido a la alta complejidad y el alto alcance requerido para desarrollar la idea. No se veía viable para un solo desarrollador conseguir un resultado decente.

Posteriormente, se descartó también la primera idea. A pesar de que la idea de salas se adecuaba más a lo que se quería desarrollar, la idea de mecánica principal no daba tanto juego para desarrollar una demo completa. Además, la estética realista que se había pensado estaba también fuera del alcance por la gran carga artística requerida.

Finalmente, quedaron dos ideas. Ambas ideas eran bastante potentes para desarrollar. El alcance era moldeable, es decir, que se podrían diseñar e implementar puzzles poco a poco según el tiempo que habría disponible, y la complejidad y la carga artística era la justa. Aun así, se tenía que elegir una para seguir avanzando por lo que se eligió la originalidad como criterio decisivo. Aunque ambos conceptos eran parecidos, es cierto que la cuarta idea tenía una mecánica más interesante y original que explotar, ya que actualmente existen muchos videojuegos del estilo “Escape Room”.

Así que finalmente, se escogió la cuarta idea como base para desarrollar este proyecto.

3.3.2 GDD

El siguiente paso para seguir ha sido la creación del Game Design Document, o GDD. El GDD es un documento de suma importancia en un desarrollo de videojuegos. En él, se recopila toda la información del desarrollo y sirve principalmente para ayudar en la comunicación de los

diferentes equipos de desarrollo. Técnicamente, el GDD, es un documento de referencia el cual se debe consultar frecuentemente (Martins et al., 2019).

En este caso, al tratarse de un proyecto de un solo desarrollador, se empleó el GDD principalmente para dejar definidas todas las mecánicas del videojuego para dejar claro el alcance de éste.

De forma resumida, se trata de un videojuego de puzzles en primera persona cuya mecánica principal es recoger y pintar elementos. Además, está dividido en diferentes salas. En el videojuego existen tres colores principales: el rojo, el verde y el azul.

- El rojo sirve para cerrar mecanismos.
- El verde sirve para abrir mecanismos.
- El azul sirve para que elementos con forma triangular se muevan.

También existen otros colores como el negro, que envía al jugador al último punto de guardado y el blanco y gris que no tienen propiedades, pero sirven para diferenciar lo que es pared de lo que son puertas. Existen elementos interactuables como puertas, puertas interconectadas y plataformas móviles.

3.3.3 Creación del proyecto

Tras dejar claro la base del diseño del videojuego, se creó el proyecto. El motor de videojuegos empleado ha sido Unity. Unity es un motor de videojuegos muy potente, especialmente suele ser el motor que elegir para desarrollos independientes por la gran cantidad de tutoriales y comunidad que existe hasta hoy día.

El proyecto de Unity se creó con la versión 2022.3.16f1. Esta versión es la última versión estable o LTS. Es recomendable utilizar versiones estables porque, como su nombre indica, es una versión que es poco probable que genere errores críticos. Además, se eligió la plantilla Universal Render Pipeline (URP) (Fig. 15). Esto es así porque se pretende incluir un shader, o sombreador, para la estética del videojuego. Esta plantilla incluye herramientas y configuraciones que nos facilita a la hora de crear estos assets.

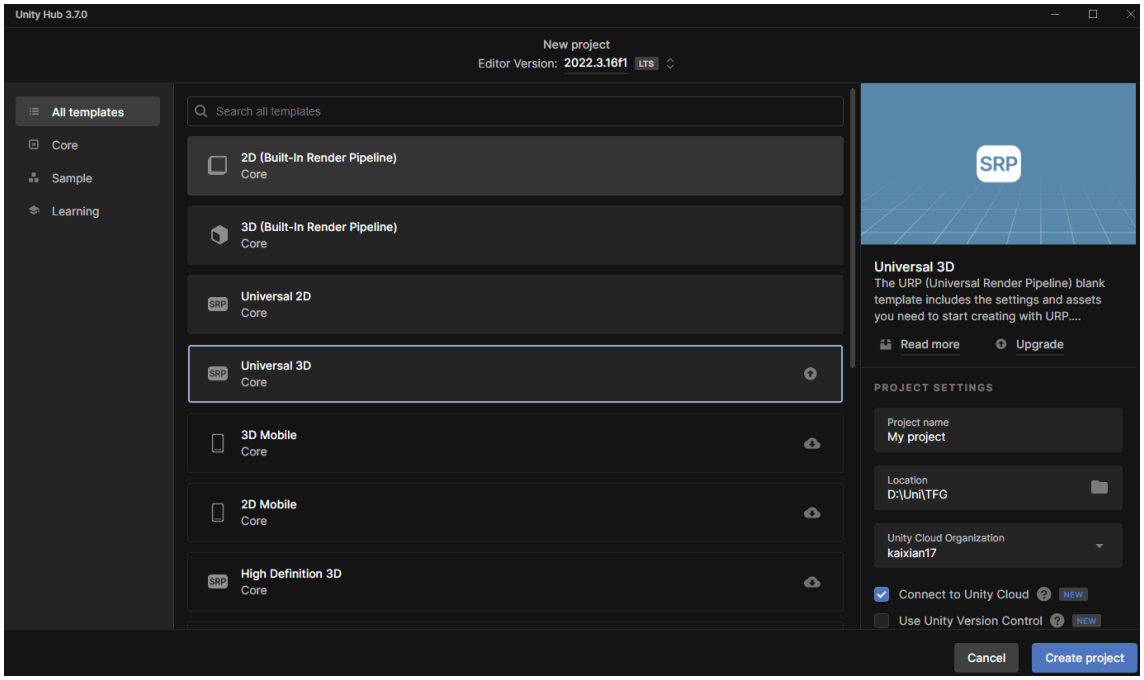


Figura 15. Captura de la plantilla empleada para el proyecto. (Fuente: elaboración propia)

Por otro lado, también se creó un proyecto en git, específicamente en Github. Git es un software de control de versiones, es decir, que permite al usuario tener control sobre los cambios que está subiendo a este software. En este caso, se empleó Github y su aplicación de escritorio. Github es una plataforma, de entre muchas, para alojar proyectos y utiliza el sistema de control de versiones git. Se ha elegido esta por la comodidad de conocerla de trabajos previos.

Tras crear el proyecto se subió el proyecto de Unity al proyecto de Github (o también llamado repositorio) (Fig. 16).

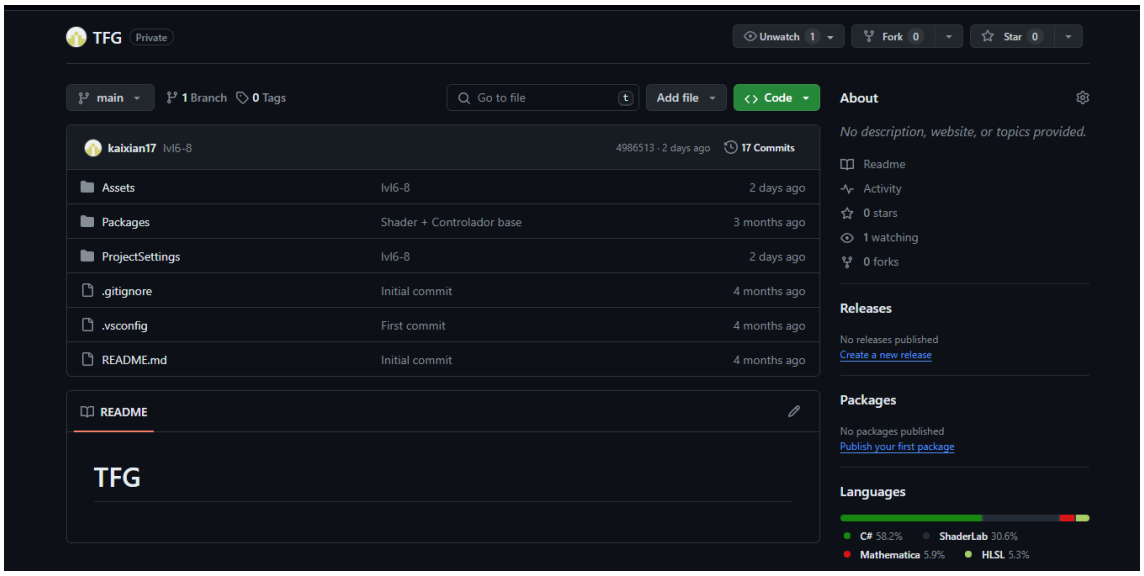


Figura 16. Captura del repositorio del proyecto. (Fuente: elaboración propia)

3.3.4 Búsqueda de tutoriales

Para el desarrollo del videojuego, ha sido de gran ayuda la búsqueda de algunos tutoriales. Estos tutoriales han servido para sentar las bases del proyecto. Principalmente se han buscado dos tipos de tutoriales:

- Tutoriales de controlador en primera persona.
  - https://www.youtube.com/watch?v=f473C43s8nE
  - https://www.youtube.com/watch?v=LqnPeqoJRFY

- Tutoriales de shader de outline en URP.
  - https://www.youtube.com/watch?v=74AS5DmLe8w
  - https://www.youtube.com/watch?v=VGEz8oKyMpY
  - https://www.youtube.com/watch?v=LMqio9NsqmM

3.3.5 Organización y reuniones (Metodología Scrumban)

Respecto a la organización del proyecto, aunque el desarrollo conste de un solo integrante, ha sido necesario crear un plan de proyecto para seguir adelante. En este caso, se ha empleado la herramienta HacknPlan, la cual es una herramienta que permite al usuario crear tareas y dividirlas en hitos (en resumidas cuentas, sigue la metodología Kanban).

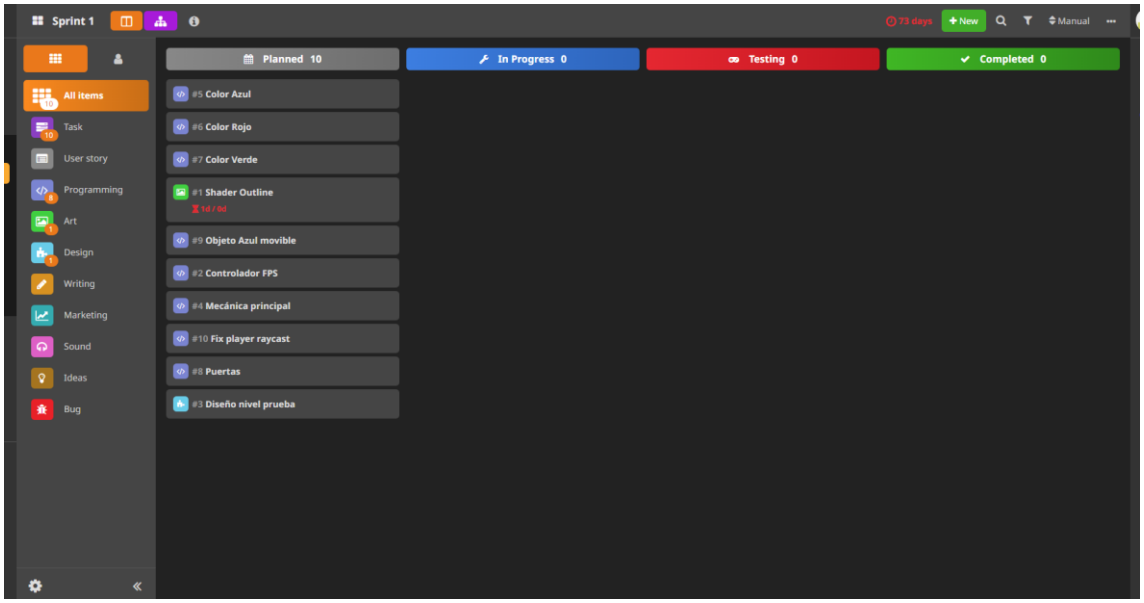


Figura 17. Captura del tablón de HacknPlan del proyecto. (Fuente: elaboración propia)

A parte del tablón de tareas, durante el desarrollo hubo algún que otro problema con los tiempos. Por ello, junto al tutor de TFG, se planteó un plan de producción basado en pequeñas reuniones y sprints de una duración de 1-2 semanas. En estas reuniones se enseñaba el avance del producto al tutor, quien daba feedback, y posteriormente se planificaba qué habría que enseñarle en la siguiente reunión. Además, si sumamos ambas metodologías conseguiríamos

una metodología Scrumban, la cual es una de las más empleadas en el sector de desarrollo de software. Con esto, se consiguió una mejora notable en la productividad.

3.4 Producción

3.4.1 MVP

El primer paso que se realizó en esta etapa fue desarrollar el MVP, o Minimun Viable Product. El MVP es un producto que contiene lo mínimo para que funcione el videojuego. Se suele utilizar como punto de parada para ver si se continúa o no el desarrollo. Esto es así debido a que realizar un MVP no sólo consigue información sobre si el producto funciona, sino que ahorra muchísimos recursos de desarrollos innecesarios a un estudio o empresa de videojuegos (Hyrynsalmi et al., 2018).

Controlador base

Lo primero que se implementó fue el controlador base del jugador. En este caso se eligió un tutorial que ya se había empleado en anteriores proyectos de un desarrollador llamado Dave. Este controlador contiene movimiento básico en primera persona, lo justo y necesario para utilizarlo como base para el proyecto.

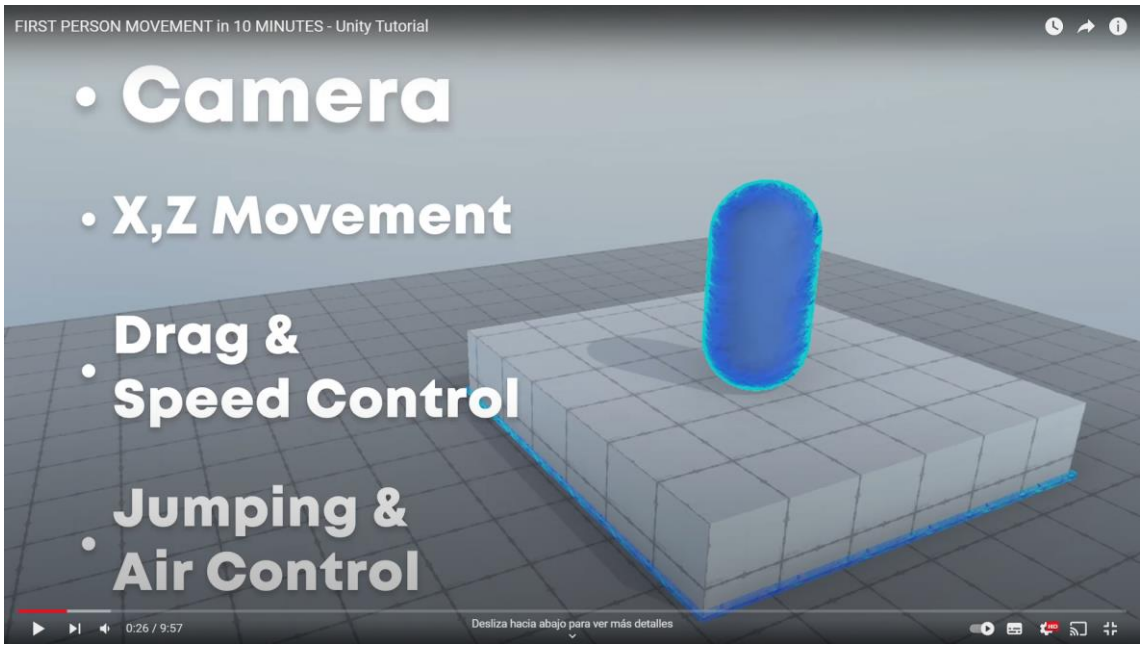


Figura 18. Captura de características que tiene el controlador base (Fuente: extraído de https://www.youtube.com/watch?v=f473C43s8nE)

Shader

Una vez implementado el controlador en primera persona, se implementó el shader. Para este videojuego se quería implementar un shader que marcase de negro los contornos de los objetos de escena (o también llamado "outlines").

Para ello, primero se agregaron materiales "unlit" (materiales que no le afectan la luz, por lo que no emiten sombras) a los elementos de la escena (Fig. 19).

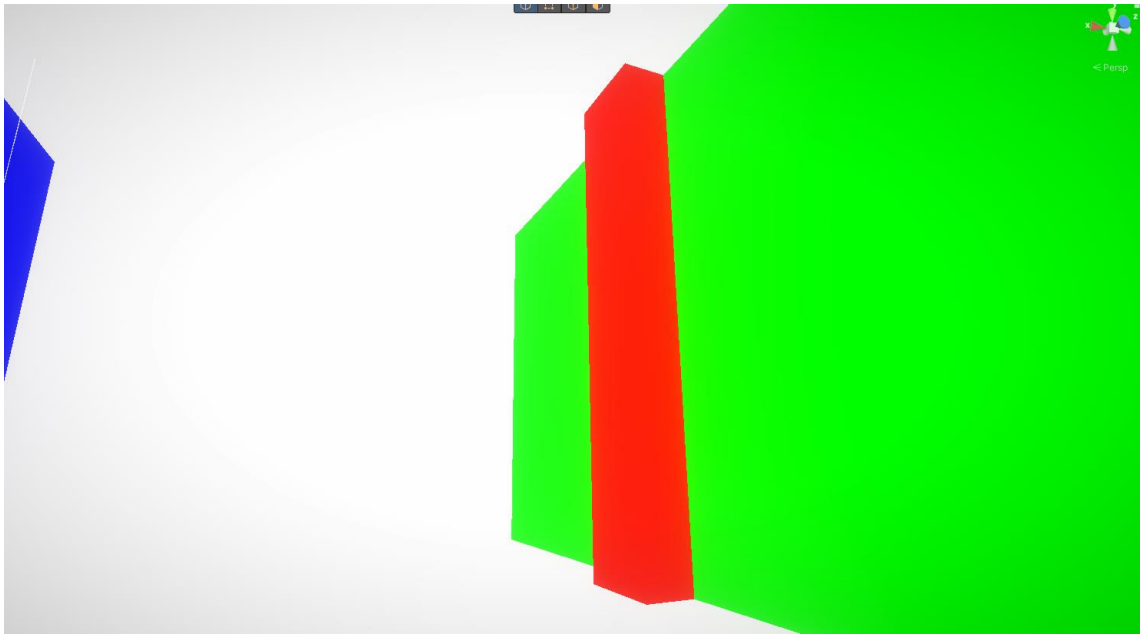


Figura 19. Materiales unlit aplicados a elementos de la escena. (Fuente: elaboración propia)

Entonces, teniendo como apoyo los tres vídeos encontrados previamente se procedió a implementar el shader utilizado la herramienta "ShaderGraph" de Unity (Fig. 20). Lo importante para que este tipo de shader funcione es que la cámara consiga diferenciar los objetos de la escena. Para ello, ha sido necesario agregar al shader tres parámetros que hagan posible esto. Los parámetros son: el color, las normales y la profundidad. Con estos tres parámetros, se puede ajustar el shader para que funcione como se desee.

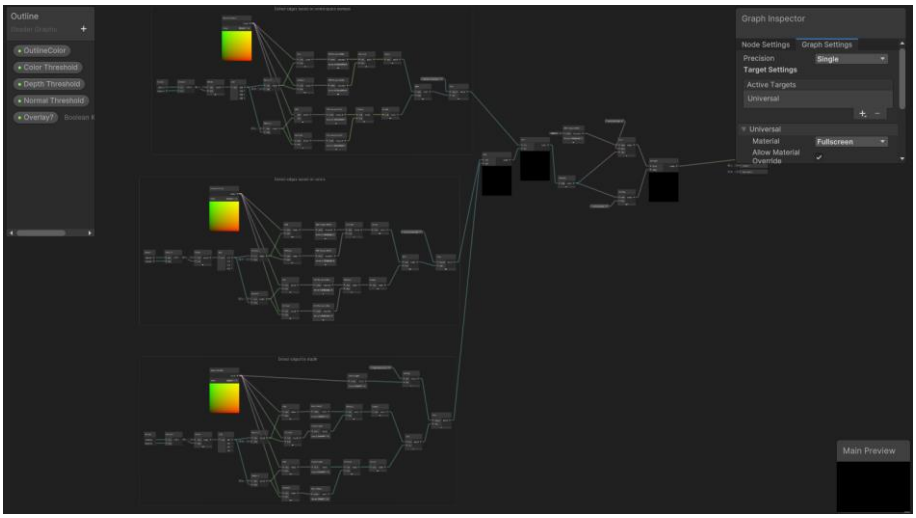


Figura 20. Shader de outlines empleado para el proyecto. (Fuente: elaboración propia)

Una vez creado el shader, queda implementarlo. Para ello, se agregó un pase de render, es decir, una capa a la cámara, en los ajustes de URP (Fig. 21).

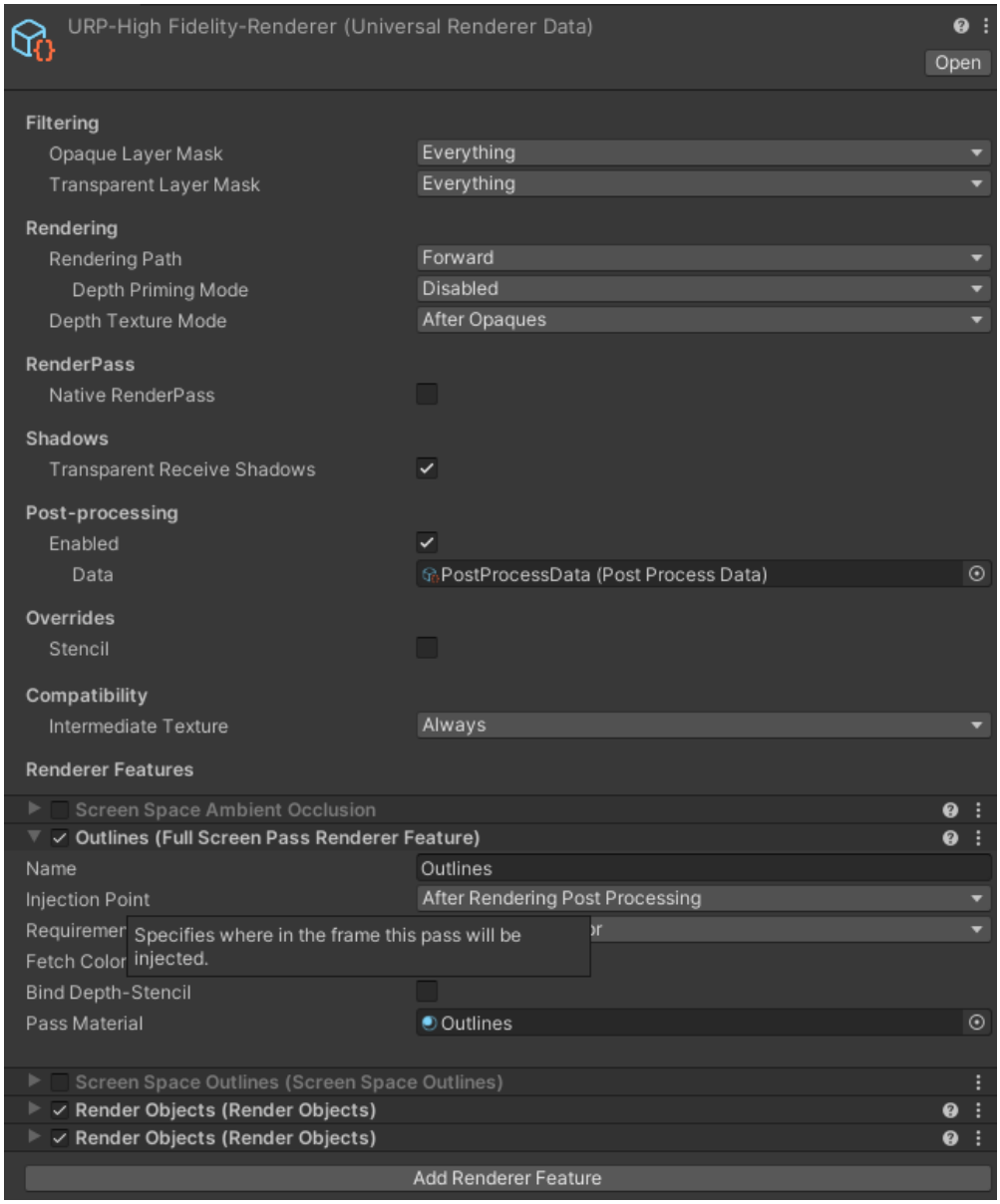


Figura 21. Ajustes de URP. (Fuente: elaboración propia)

Finalmente, tras ajustar un poco los parámetros se consiguió el siguiente resultado:

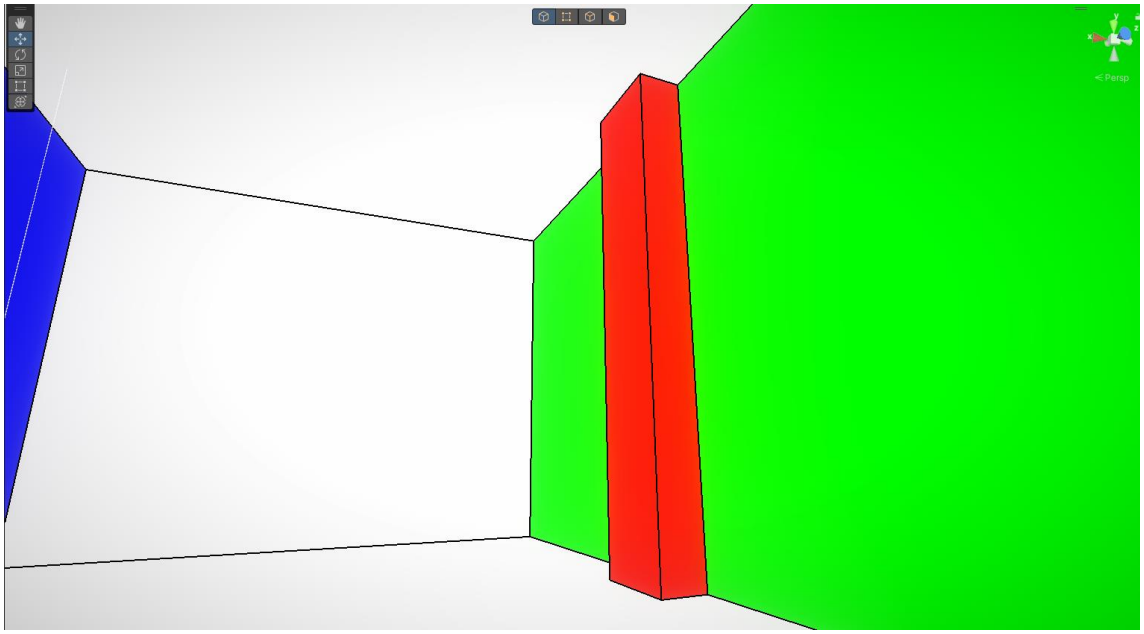


Figura 22. Resultado del shader implementado. (Fuente: elaboración propia)

Mecánica base

Por consiguiente, se implementó la mecánica principal del videojuego, recoger y pintar objetos. Lo primero que se implemento fue crear el cambio de color. Para ello, se añadió la librería DoTween al proyecto. Esta librería contiene diferentes funciones que hacen transiciones homogéneas y limpias. En este caso se empleó la función DOColor() (Fig. 23), la cual permite hacer una transición entre dos colores. Cabe destacar que esto se encuentra en un script que se ha incluido en cada objeto que se puede pintar.

```
1 referencia
IEnumerator ColorFade(Color color)
{
    isColorChangeActive = false;

    gameObject.GetComponent<Renderer>().material.DOColor(Color.gray, baseToBlackTime);

    yield return new WaitForSeconds(baseToBlackTime);

    gameObject.GetComponent<Renderer>().material.DOColor(color, blackToColorTime);

    yield return new WaitForSeconds(blackToColorTime);

    isColorChangeActive = true;
}
```

Figura 23. Función cambio de color ColorFade(). (Fuente: elaboración propia)

Luego, se agregó la función de poder pintar objetos en escena. Esto funciona prácticamente igual que la función de disparar en un videojuego de disparos en primera persona. Se lanza un

rayo desde la cámara (un “raycast”), y si golpea algo que contenga el script del cambio de color, se ejecuta la función del cambio de color (Fig.24).

```
RaycastHit hit;

if (Input.GetKeyDown(shoot))
{
    if (Physics.Raycast(cam.transform.position, cam.transform.forward, out hit, shootRange))
    {

        if (hit.collider.GetComponent<ColorObject>() != null && !hit.collider.GetComponent<ColorObject>().isColorStatic)
        {
            hit.collider.GetComponent<ColorObject>().ColorChange(selectedColor);
            hit.collider.tag = selectedColorName;
        }
```

Figura 24. Disparo o función de pintar del controlador. (Fuente: elaboración propia)

Finalmente, se añadió la función de recoger colores, ya que actualmente solo funciona pintar de un solo color. Esto funciona prácticamente igual que el disparo. Lo que cambia se es que, en vez de cambiar información, la recoge (Fig. 25). En este caso se recoge el color y la etiqueta o tag que servirá para las propiedades de los colores.

```
if (Input.GetKeyDown(dropper))
{
    if (Physics.Raycast(cam.transform.position, cam.transform.forward, out hit, dropperRange))
    {
        if (hit.collider.GetComponent<ColorObject>() != null && hit.collider.GetComponent<ColorObject>().isColorChangeActive)
        {
            selectedColor = hit.collider.GetComponent<Renderer>().material.color;
            selectedColorName = hit.collider.tag;
```

Figura 25. Función de recoger colores. (Fuente: elaboración propia)

Puerta normal (Colores verde y rojo)

Una vez metida la mecánica principal, se ha comenzado a implementar las propiedades de los colores. Los primeros colores para introducir han sido el rojo y el verde para las puertas. Las puertas incluyen dos elementos principales: la puerta, que tiene asociada una animación de abrir y cerrar; y un interruptor (en forma de cubo), que sirve para abrir o cerrar el mecanismo (Fig. 26).

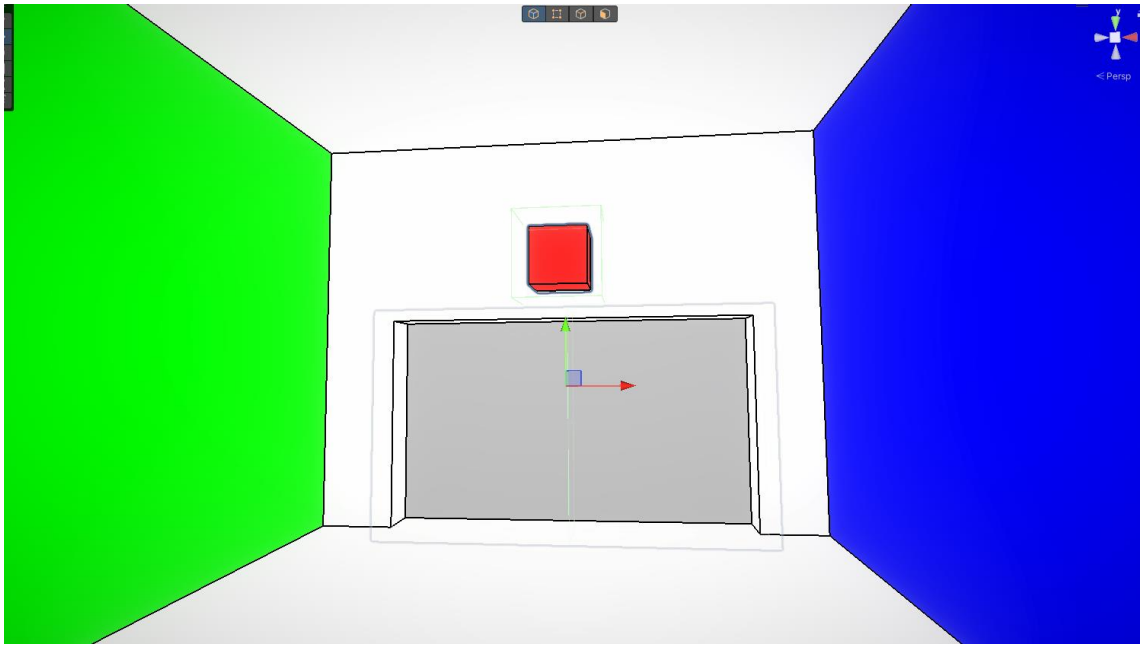


Figura 26. Modelo de puerta dentro del proyecto. (Fuente: elaboración propia)

El funcionamiento es muy sencillo. La puerta está ligada a una animación con estados. Si el interruptor está en verde, el estado cambia a “Open”, en cambio si está en rojo, cambia a “Close” (Fig. 27).

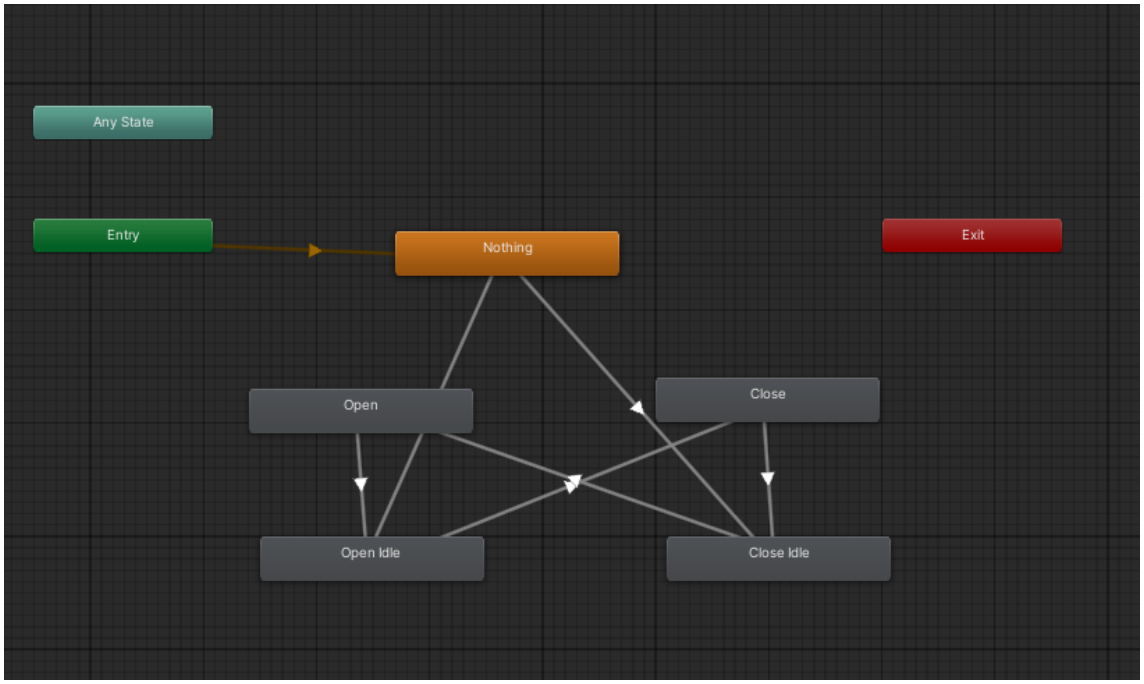


Figura 27. Sistema de animaciones de la puerta. (Fuente: elaboración propia)

Puerta interconectada

Para el proyecto, también ha sido necesario implementar puertas interconectadas entre sí. El funcionamiento es bastante simple. Si una puerta está abierta, la otra se encontrará cerrada, y así viceversa (Fig. 28).

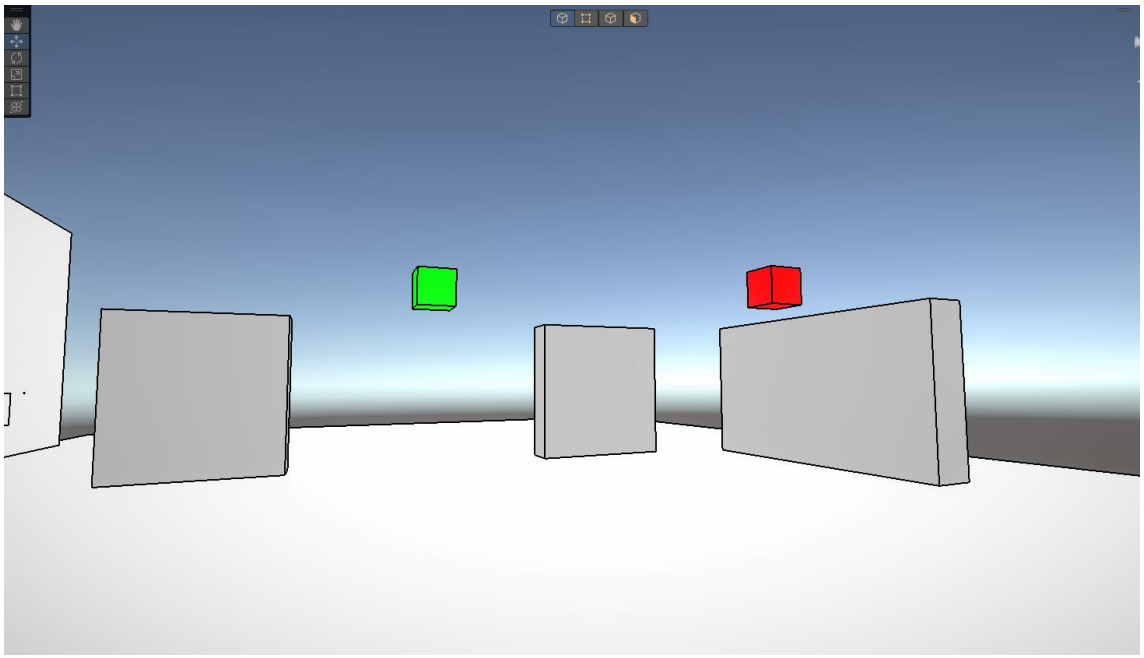


Figura 28. Puertas interconectadas. (Fuente: elaboración propia)

Color Azul y negro

Para acabar, se ha implementado las propiedades del color azul y negro. En cuanto al azul, son plataformas que siguen un camino fijado. Los puntos del camino se encuentran dentro de un objeto que lleva la lógica del movimiento. Asimismo, la plataforma solo se moverá si el color del objeto es azul, si es otro color se parará (Fig. 29).

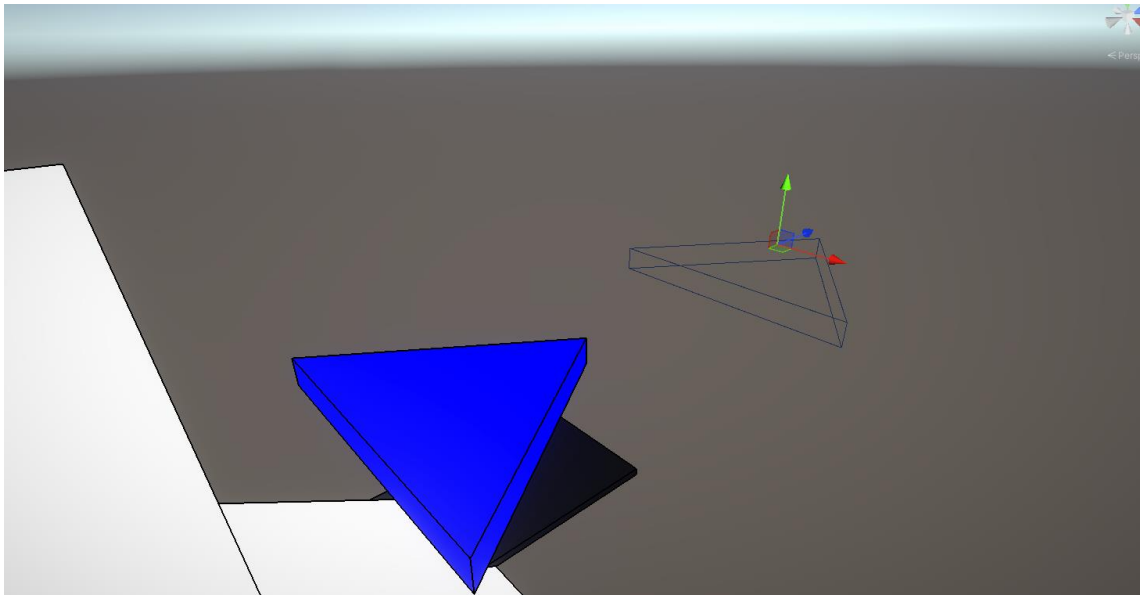


Figura 29. Plataforma móvil y su siguiente posición. (Fuente: elaboración propia)

Por otro lado, al añadir plataformas móviles, el jugador puede caerse. Por ello, se ha añadido propiedades de reaparición al color negro. El funcionamiento es muy sencillo. Se trata de un trigger ligado a un punto de reaparición. Si el jugador toca el trigger, se teletransportará al punto de reaparición designado (Fig. 30).

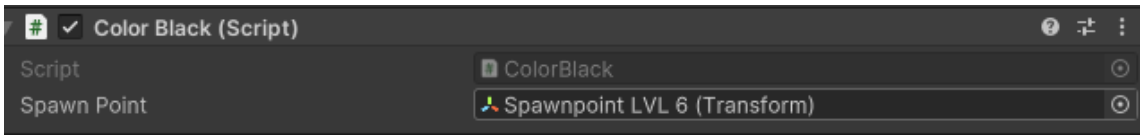


Figura 30. Componente del funcionamiento del color negro. (Fuente: elaboración propia)

Tras implementar todo lo esencial para el funcionamiento del videojuego, se completó el MVP.

3.4.2 Diseño de niveles

El siguiente paso fue diseñar los niveles. El proyecto está dividido en diferentes salas. Gracias a esto, es posible no tener un número fijo de niveles porque siempre es posible agregar más. Aun así, en reuniones con el tutor, se acordó implementar 10 salas en el videojuego.

Los diseños de los niveles se realizaron en Photoshop siguiendo la siguiente leyenda:

Figura 31. Leyenda de diseño de niveles. (Fuente: elaboración propia)

Por ejemplo, en la Figura 32, se puede observar el diseño del nivel 4. Principalmente son salas cuadradas y cada uno de los diseños tiene el orden de las acciones mostrada. Si en algún caso, hay algún elemento que no se encuentra en la leyenda, se informará en uno de los laterales. En este caso, en el nivel hay ventanas y está marcado en la parte inferior del diseño. Asimismo, también queda reflejado el número total de acciones necesarias para completar el nivel. Esto se ha implementado siguiendo la idea del estudio de Linehan y sus coinvestigadores del Portal 1 y 2.

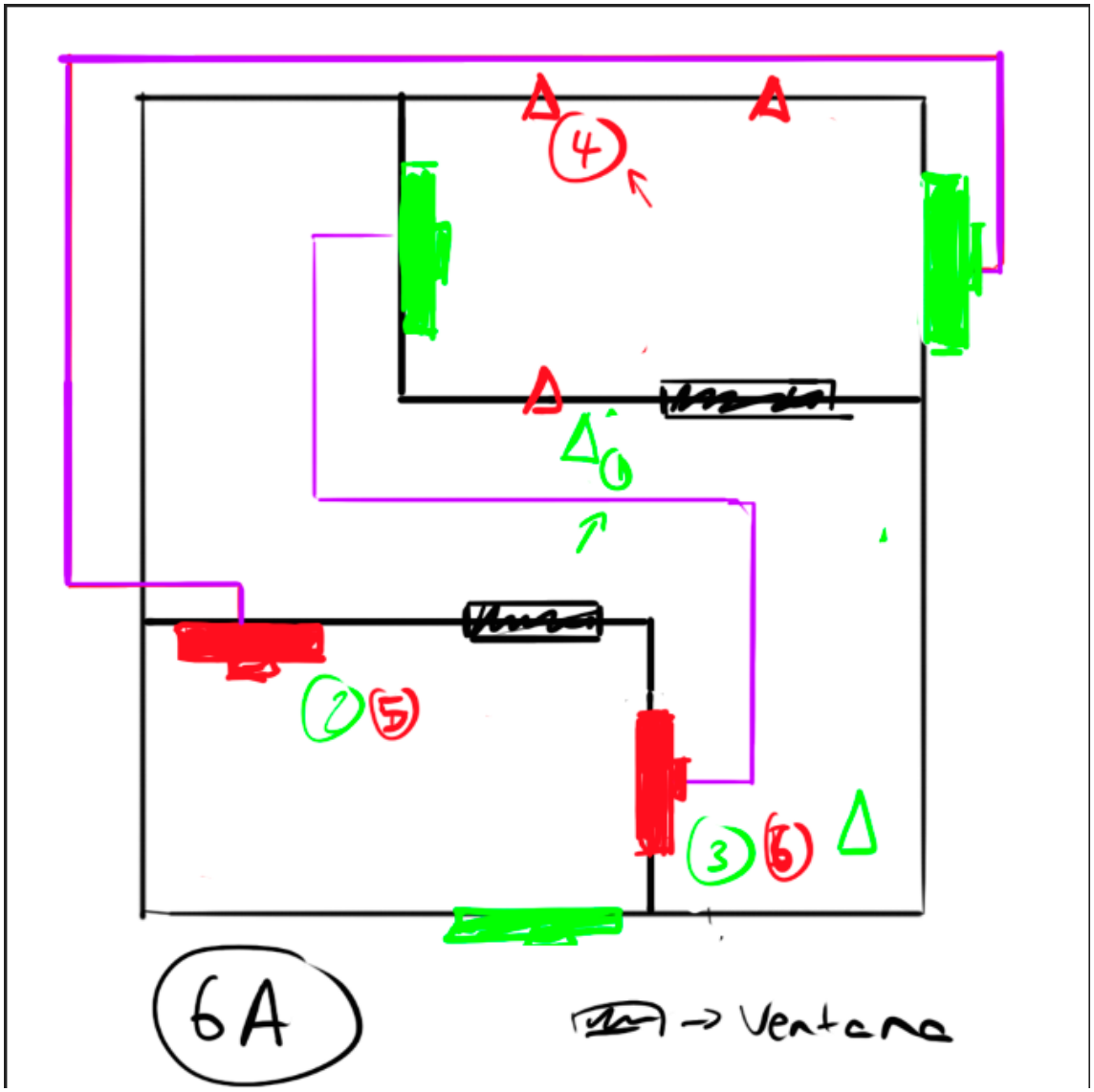


Figura 32. Diseño del nivel 4. (Fuente: elaboración propia)

En total, contando con el nivel 0, se diseñaron los 10 niveles acordados (Fig. 33).

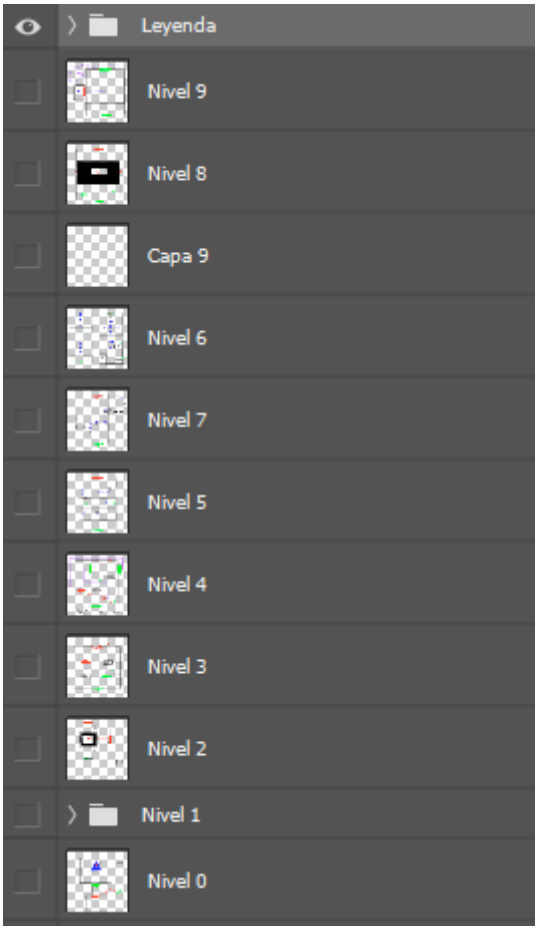


Figura 33. Desglose de los diseños de niveles en Photoshop. (Fuente: elaboración propia)

3.4.3 Modelo de la pistola

A continuación, antes de implementar los niveles ya diseñados, se quiso modelar el arma. El programa que se empleó fue Blender. La idea con esta arma es conseguir algo relacionado con los colores, debido a la naturaleza del proyecto. Por ello, se decidió hacer una pistola de paintball. Sin embargo, al modelarla, como el cañón de estas pistolas es extremadamente larga, no concordaba bien con el género de puzzles. Más bien parecía un arma para un videojuego de disparos. Por ello, lo que se hizo fue recortar el cañón y hacerlo más grueso obteniendo el siguiente resultado (Fig. 34):

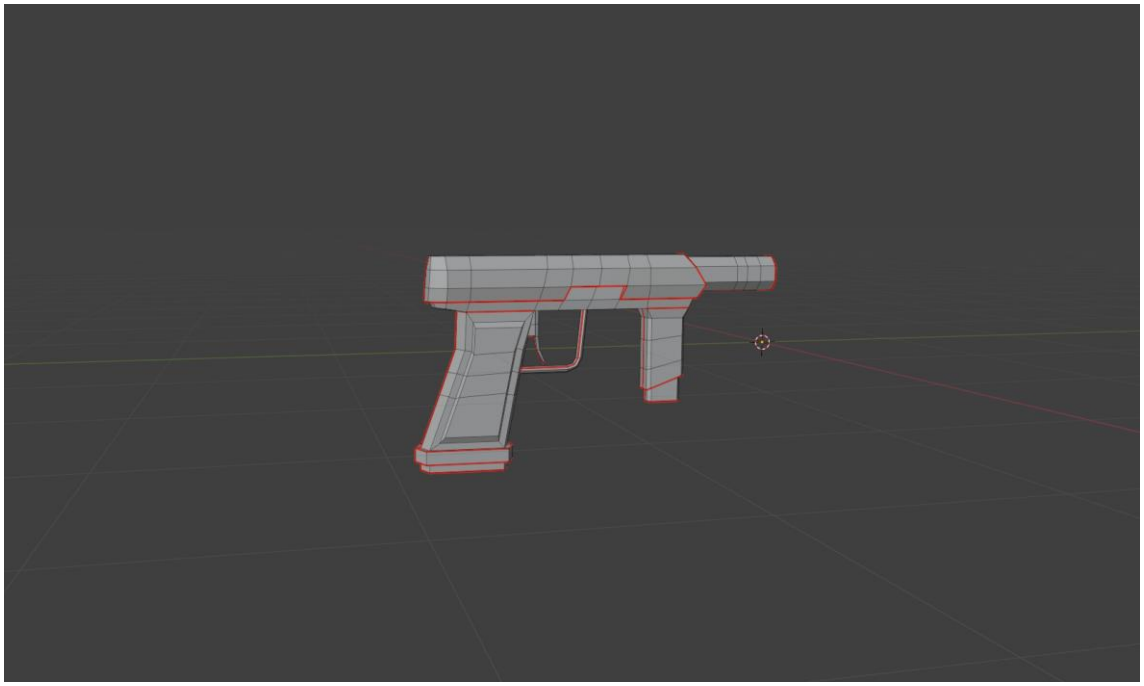


Figura 34. Modelo de arma en Blender. (Fuente: elaboración propia)

Luego, se exportó el modelo en formato .fbx, y se implementó en el proyecto. No obstante, quedaba algo raro, ya que al recoger el color no había ningún tipo de feedback que indicase qué color tenía guardada el arma (Fig. 35).

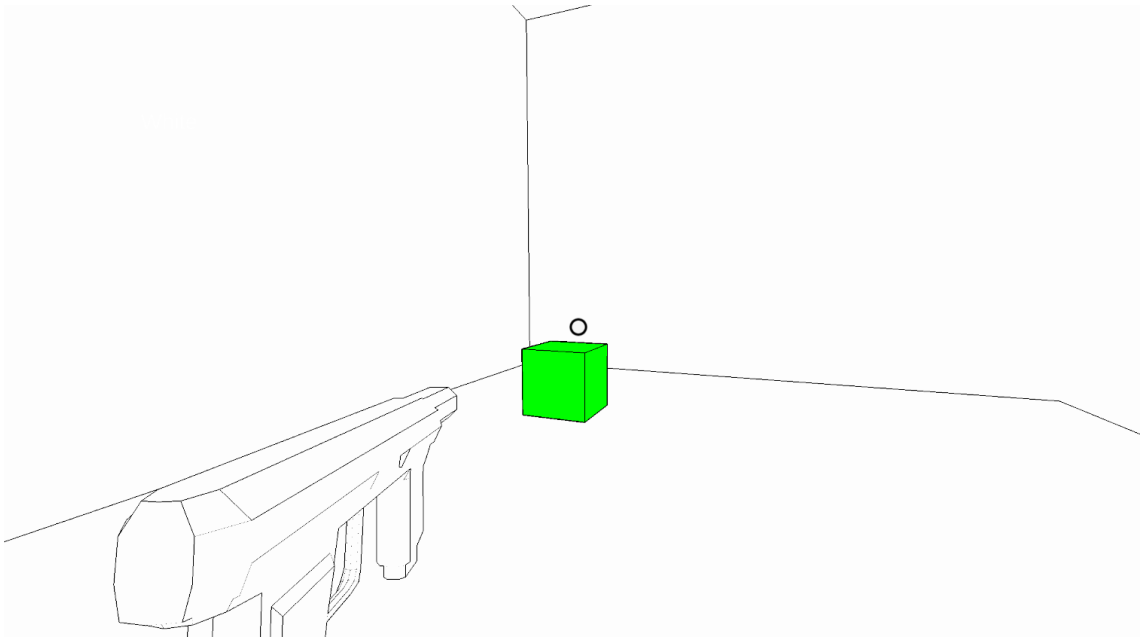


Figura 35. Modelo del arma implementada en el motor. (Fuente: elaboración propia)

Por ello, se agregó feedback visual al arma. Lo que se implementó fue que, al recoger los colores, el arma se pintaba de ese color, dando feedback visual al jugador en todo momento de qué color tenía guardado (Fig. 36). Además, se eligió esta forma porque es una manera diegética de implementar este tipo de feedback en vez de un texto flotante en la pantalla.

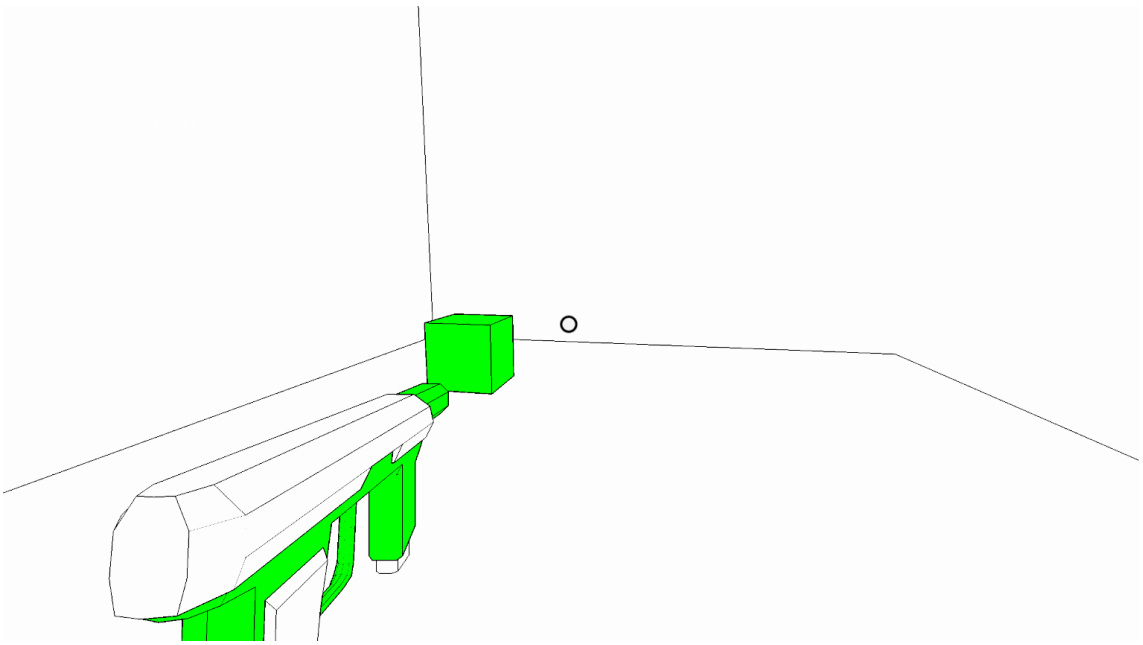


Figura 36. Feedback visual pintado en el arma. (Fuente: elaboración propia)

3.4.4 Implementación de niveles

Tras modelar e implementar el arma, se comenzó la implementación de los niveles. Para ello, fue necesario una herramienta que ahorró muchísimo tiempo a la hora de implementar los niveles, Probuilder. Esta herramienta sirve para poder modelar dentro de Unity, y principalmente sirve para prototipar con formas básicas escenarios y niveles dentro de los videojuegos. Como el proyecto que se está desarrollando utiliza una estética muy simple con objetos geométricos, esta herramienta vino muy bien para implementar los niveles

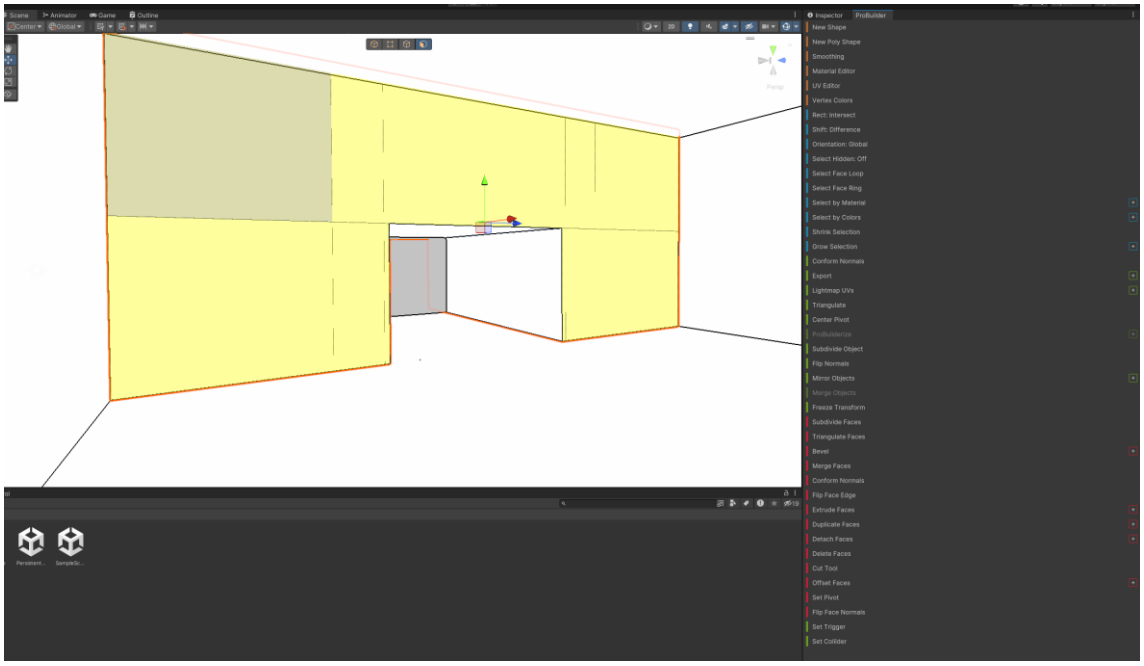


Figura 37. Herramienta Probuilder para la implementación de niveles. (Fuente: elaboración propia)

Con esta herramienta y con paciencia, se ha ido introduciendo cada uno de los niveles diseñados previamente. A continuación, se pondrá el nivel 4, que se puso de ejemplo anteriormente, implementado:

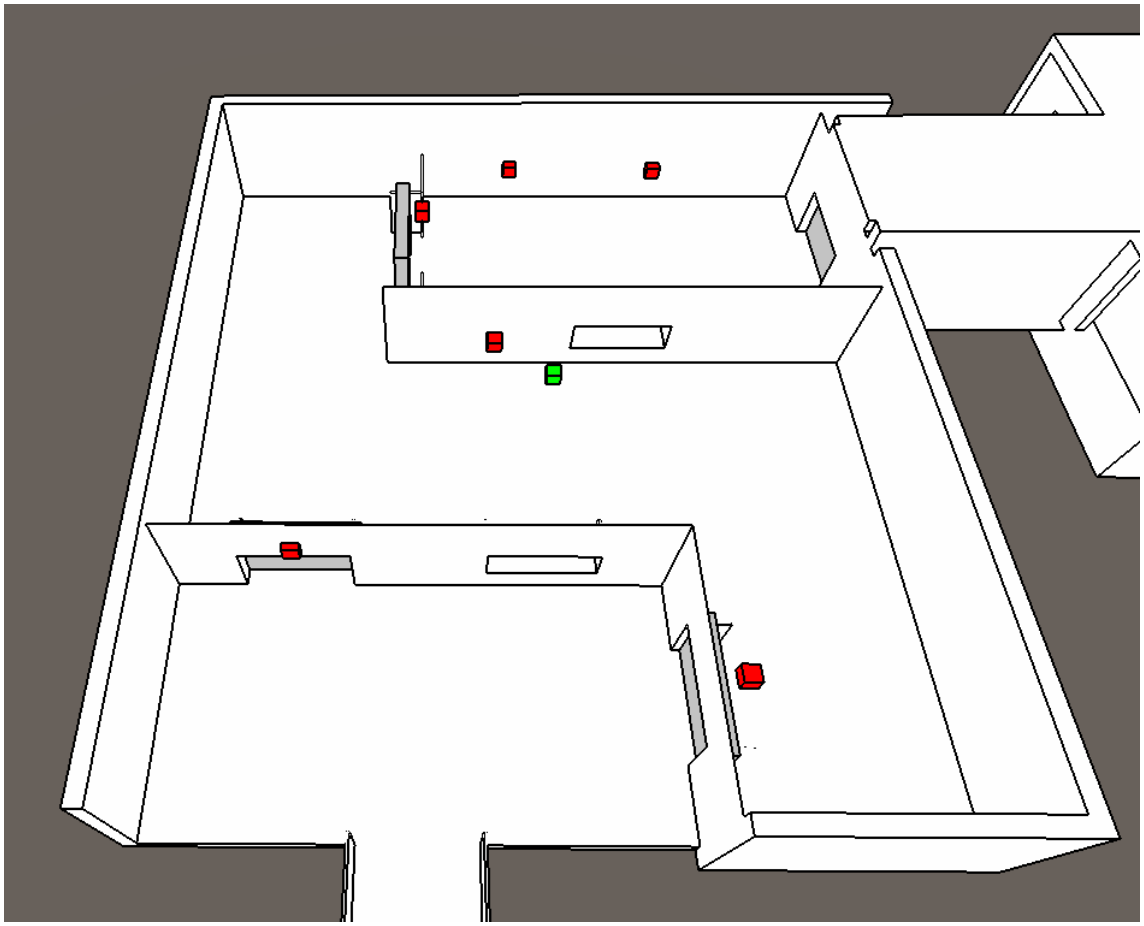


Figura 38. Nivel 4 implementado (Fuente: elaboración propia)

Finalmente, tras implementar todos los niveles, se obtuvo el siguiente resultado (Fig. 39), donde todos los niveles están conectados entre sí.

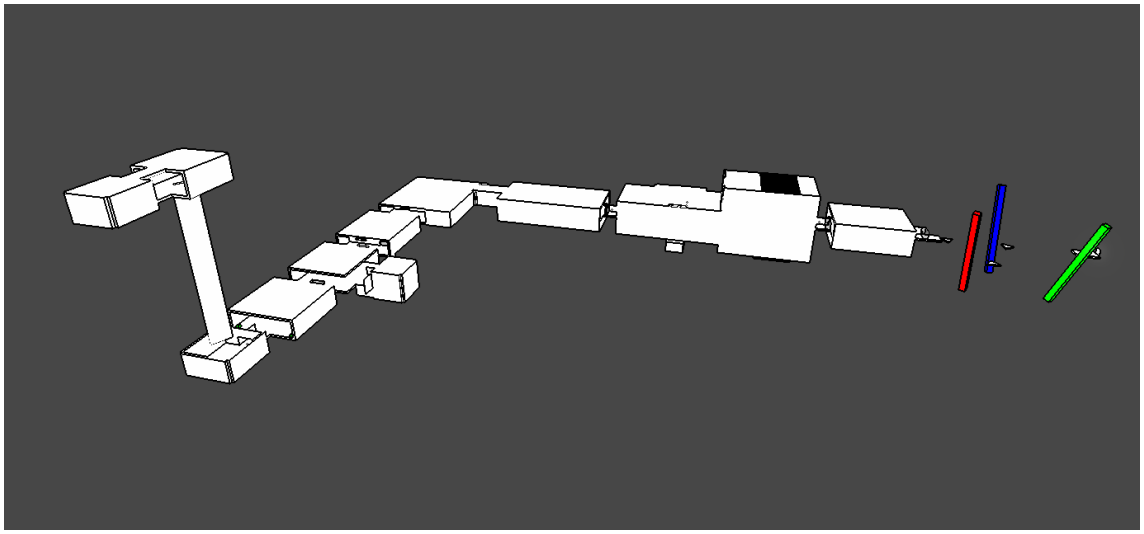


Figura 39. Captura de todos los niveles implementados y conectados. (Fuente: elaboración propia)

3.4.5 Interfaz de usuario

El siguiente paso de este desarrollo fue incluir el apartado de interfaz de usuario, o UI. Para el diseño de la interfaz se tuvo en cuenta la propia temática del videojuego, obteniendo el siguiente resultado:

Figura 40. Resultado del menú principal. (Fuente: elaboración propia)

En el ejemplo de la Figura 40, se puede observar el menú principal. Siguiendo la línea del proyecto, se decantó por una interfaz sencilla. Se utilizaron colores sólidos que están todo el rato presente en el videojuego, el negro y blanco. Además, se empleó una fuente Serif. Esto es así por la esencia de misterio y abstracción similar al videojuego analizado The Witness (Thekla, Inc., 2016). Con el menú de pausa se realizó algo similar que en el menú principal.

Por otro lado, también se agregó un menú de opciones simple con configuración de audio y sensibilidad.

Finalmente, se añadieron elementos de feedback, para que el jugador sepa en todo momento qué está haciendo. Para ello se agregaron dos características: un pequeño “hover”, es decir, cuando el cursor está por encima de un botón; y efectos de sonidos.

3.4.6 SFX

Una vez terminado de implementar todo lo que es el videojuego se decidió incluir el apartado de sonido. Lo primero a realizar ha sido un listado de sonidos a implementar. La lista es la siguiente:

- SFX de pasos
- SFX de recoger color
- SFX de disparo
- SFX de limpieza de color

- SFX de plataforma movible
- SFX de puerta abriendo
- SFX de puerta cerrando
- SFX de botones
- SFX de puzzle completado
- SFX ambiental de fondo

El siguiente paso fue conseguir los sonidos. Como hacer los sonidos era una meta inalcanzable debido a el tiempo disponible y a las capacidades personales, se decidió buscar librerías. Entonces se realizó una búsqueda por la tienda de assets de Unity, por los assets propios de sonido que se habían comprado previamente, y también sonidos extras que se habían utilizado en proyectos previos.

Como resultado, se logró reunir los sonidos necesarios. Para los sonidos que ocurren dentro de la escena de juego, se emplearon sonidos del pack "*Magic Spells Sound Effects Pack*". Este pack incluye un total de 340 sonidos con los que jugar. Luego, para los sonidos de la interfaz de usuario, se emplearon sonidos de un asset gratuito llamado "*UI Sound Effects Collection Pack 2: Buttons*". Finalmente, para el sonido de pasos, se utilizaron los sonidos de pasos que se habían empleado en un proyecto anterior.

3.5 Postproducción

3.5.1 Testeo y pulido

En esta última etapa, se ha realizado una fase de testeo y pruebas. En ella, se ha enviado a diferentes personas (amigos, familiares, profesores y conocidos) para que puedan probar el videojuego. Esta etapa es una de las más importantes en un desarrollo de videojuegos. Esto es así porque se simula lo que vivirán los jugadores que jueguen al videojuego que se ha estado desarrollando.

Además, gracias a esta fase de testeo, es posible recoger comentarios y opiniones de feedback para ver posibles fallos, mejoras y sensaciones en general. De las 20 personas, se ha sacado la siguiente lista de cosas a mejorar:

- La mecánica de las puertas dobles no se entiende.
- Añadir checkpoints o puntos de reaparición.
- Hay algún que otro pico de dificultad (Nivel de plataformas cayendo)

Una vez obtenido la lista, llegó el momento de sacar soluciones. Para el primer elemento, simplemente se añadieron cables que conectan estas puertas (Fig. 41). Es una forma de enseñar al jugador que hay dos elementos interconectados, y se había pasado por alto al implementar las puertas.

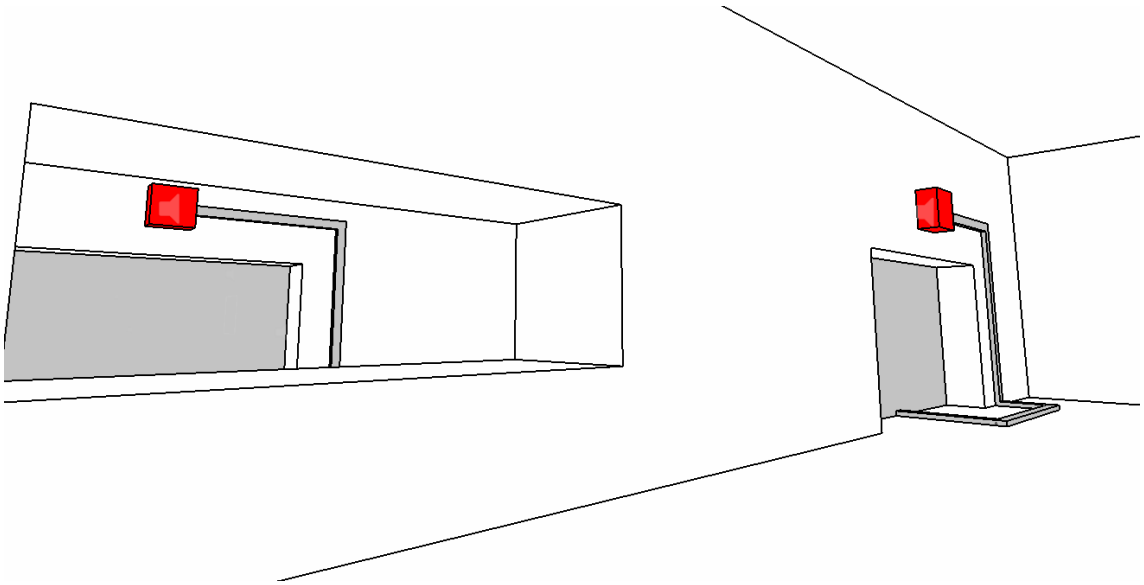


Figura 41. Solución al problema de confusión de puertas interconectadas. (Fuente: elaboración propia)

Para el segundo, se agregaron triggers, o lanzadores de eventos, estratégicos por los niveles. Esto ayuda a los jugadores a no tener que rehacer partes de puzzles que ya habían realizado. Es algo que es muy común en todo tipo de videojuegos.

Asimismo, para ampliar este sistema, se implementó un sistema de serialización o de guardado. Para ello, se utilizó una librería muy común y popular llamada JSON. Esta librería permite a los desarrolladores guardar y cargar ficheros de textos con información de la partida. Así, la información de la partida no se pierde al cerrar el videojuego. Los parámetros que se guardaron fueron los puntos de reaparición y si se ha completado el tutorial o no. Además, también se aprovechó este sistema para guardar los datos del menú de opciones.

Finalmente, para el último punto, se identificó que había varios saltos muy grandes de dificultad entre el nivel 6 y el 7. Para solucionarlo, se movieron los niveles de tal forma que el nivel 7 pasó a ser el nivel 6 y así viceversa. Esta decisión se tomó porque el nivel 7 más fácil que el 6 y tiene sentido para la curva de dificultad que los niveles vayan escalando poco a poco.

3.5.2 Ajustes de daltonismo

Durante la ideación del proyecto, se quiso poner como objetivo específico la implementación de algún ajuste de daltonismo. Esto se debe a que se quería cumplimentar un objetivo de la ODS y también por la temática del proyecto: los colores, ya que tendría sentido que este proyecto tuviese algún tipo de estos ajustes de accesibilidad.

Existen varios tipos de daltonismos, pero los principales son los siguientes:

1. Daltonismo rojo-verde
    a. Deuteranomalía. Más común. Hace que ciertos tonos de verde se vean más rojas.

b. Protanomalía. Hace que ciertos tonos de rojo se vean más verdes.
c. Protanopia y deuteranopia. Ambos tipos hacen que no se puedan diferenciar entre rojo y verde.

2. Daltonismo azul-amarillo
   a. Tritanomalía. Hace que sea difícil diferenciar el azul y el verde; y el amarillo y el rojo.
   b. Tritanopia. Hace que sea difícil diferenciar el azul y el verde; y el morado y el rojo; y el amarillo y el rosado. Colores menos brillantes.
3. Daltonismo completo
   a. Monocromacia o Acropmatopsia. No se pueden ver colores en absoluto (National Eye Institute, 2023).

La idea principal era implementar como mínimo los más comunes de daltonismo rojo-verde y azul-amarillo. Pero por suerte, se encontró un asset que viene con todos los tipos de daltonismos mencionados. Este asset externo utiliza filtros para ajustar los colores. El asset se llama “SOHNE Colorblindness” y es completamente gratuito, por lo que se implementó con siguiendo el siguiente resultado:

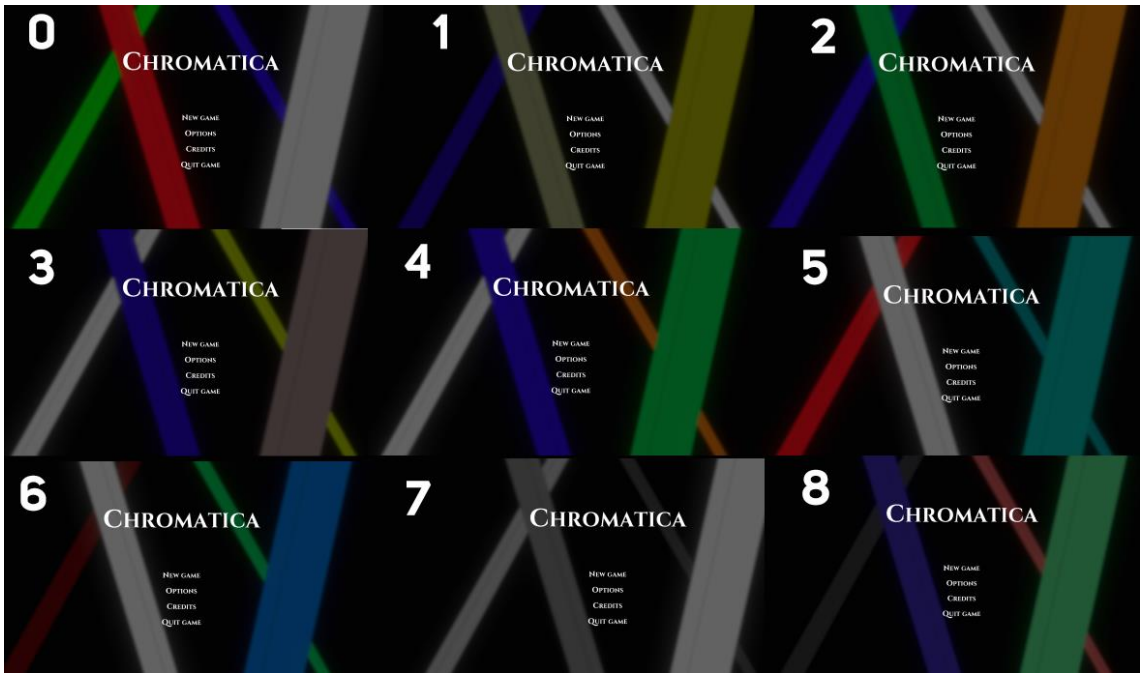


Figura 42. Ejemplos de los filtros de daltonismo implementados. (Fuente: elaboración propia)

3.6 Recursos requeridos

3.6.1 Software

- Unity
  - Dotween
  - Probuilder
  - ShaderGraph
- Github
- Visual Studio Community 2022
- HacknPlan
- Blender
- Photoshop
- Word, Excel
- Audacity

3.6.2 Tutoriales

- FIRST PERSON MOVEMENT in 10 MINUTES - Unity Tutorial
- Rigidbody FPS Controller Tutorial #1 | Basic Movement in less than 4 min
- Outlines on One Object in Unity URP Shader Graph with Edge Detection!
- Outline Post Process in Unity Shader Graph (URP)
- Outlines - Devlog 2

3.6.3 Assets utilizados

- UI Sound Effects Collection Pack 2: Buttons
  https://assetstore.unity.com/packages/audio/sound-fx/ui-sound-effects-collection-pack-2-buttons-27803

- Magic Spells Sound Effects Pack
  https://assetstore.unity.com/packages/audio/sound-fx/magic-spells-sound-effects-pack-230465

- SOHNE Colorblindness
  https://github.com/SOHNE/Colorblindness

3.7 Resultados del proyecto y análisis

3.7.1 Resultados investigación y análisis

Respecto al análisis de la tutorización en diferentes títulos de videojuegos de puzzles, se han sacado las siguientes conclusiones para tener en cuenta:

1. En los tutoriales iniciales, como mínimo, hay que mostrar el esquema de controles de las mecánicas específicas, como recoger el color y disparar para pintar en este caso. Sin embargo, otros controles más básicos como el esquema de movimiento como WASD para moverse y el ratón para mover la cámara no haría tanta falta.
2. Es importante tener en cuenta que se deben asociar elementos visuales o sonoros a mecánicas. Con ello sería más fácil para el jugador identificar qué elementos hacen qué cosas. Algunos ejemplos podrían ser el color blanco de *Portal*, que son los sitios donde poder poner portales; o en el caso de *The Witness*, cada mecánica de puzzle tiene un color y elementos asociados, y siempre es el mismo durante el transcurso del videojuego.
3. Una curva de aprendizaje y de dificultad bien construida es crucial para tutorizar cualquier juego.
4. Son bastante útiles los elementos del entorno para guiar al jugador. Un claro ejemplo que usan todos los videojuegos analizados son los cables. Esto se utiliza para señalizar mecanismos conectados.

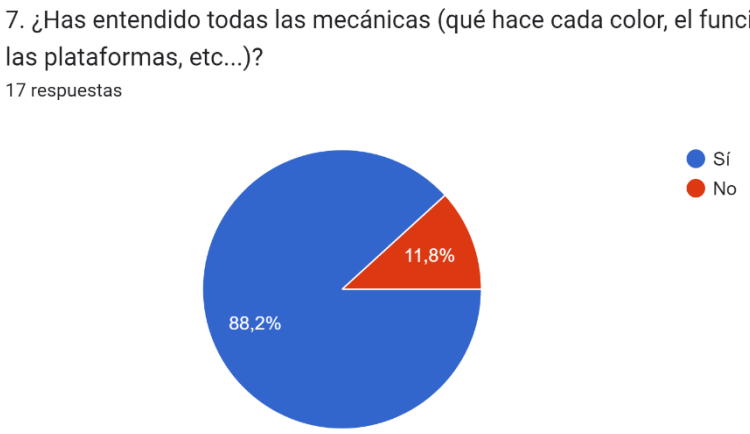
3.7.2 Resultados del proyecto y feedback

En lo que concierne a los resultados del proyecto, se ha logrado completar el objetivo de desarrollar un videojuego con elementos de tutorización implícita. Como bien se ha visto en el apartado de Metodologías, se han cumplido todos y cada uno de los objetivos específicos y sus subobjetivos.

Por otro lado, los cuatro puntos de los resultados de la investigación no solo han sido útiles para el desarrollo profesional como diseñador de videojuegos, sino que también se han implementado cada uno de ellos en el propio proyecto.

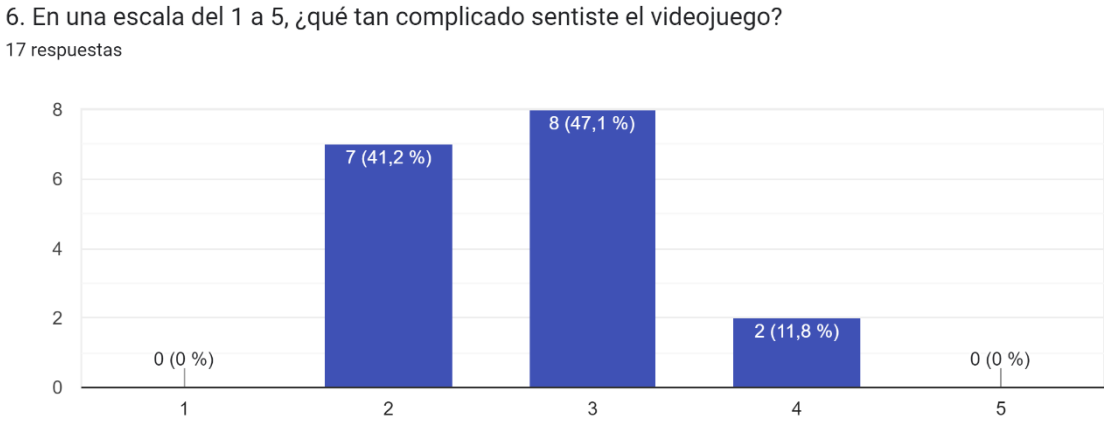
Además, se realizó un formulario para que las personas de testeo pudieran añadir algo de feedback y así conseguir datos importantes con los que mejorar el videojuego. A continuación, se mostrará un resumen por puntos de los resultados:

1. En el proyecto, se ha implementado un tutorial muy simple sin ningún elemento explícito. En el caso de este proyecto, cada color y forma tiene alguna propiedad en concreto y nunca cambia. Esto ha ayudado en que se han entendido todas las mecánicas. Como se puede ver en la Gráfica 2, solamente un 11,8% no entendió la mecánica de rebote del color rojo y ha sido porque el puzzle de introducción es de prueba y error.



Gráfica 2. Porcentaje de personas que entendieron las mecánicas. (Fuente: elaboración propia)

2. A la hora de diseñar los niveles se ha tenido mucho hincapié en que la curva de dificultad y aprendizaje sea progresiva según las mecánicas se van implementando (como bien se menciona en la gráfica de Portal). El plan era realizar un videojuego intuitivo de jugar. Se puede ver que se ha logrado, ya que la Gráfica 3 refleja que la dificultad del videojuego es normal tirando más para fácil.



Gráfica 3. Escala de dificultad del videojuego. (Fuente: elaboración propia)

3. Una de las preguntas finales fue “Comenta aquí cualquier tipo de feedback para mejorar el proyecto”. Aquí, los testers podrían dar comentarios sobre cosas a mejorar (Fig. 42). De estos comentarios, los más comunes fueron que había un salto de dificultad extra entre el nivel 7 y 8, y que el actual nivel 7 era, más que complicado, muy tedioso. Por ello, se arregló de inmediato alternando los niveles 7 y 8 en posición, y aumentando la altura de salto en el nivel 7 actual (antes del cambio de orden). Esto solucionó ambos

fallos, ya que se les preguntó específicamente si habían experimentado algo con estos dos niveles y la respuesta fue que ni se lo habían planteado.

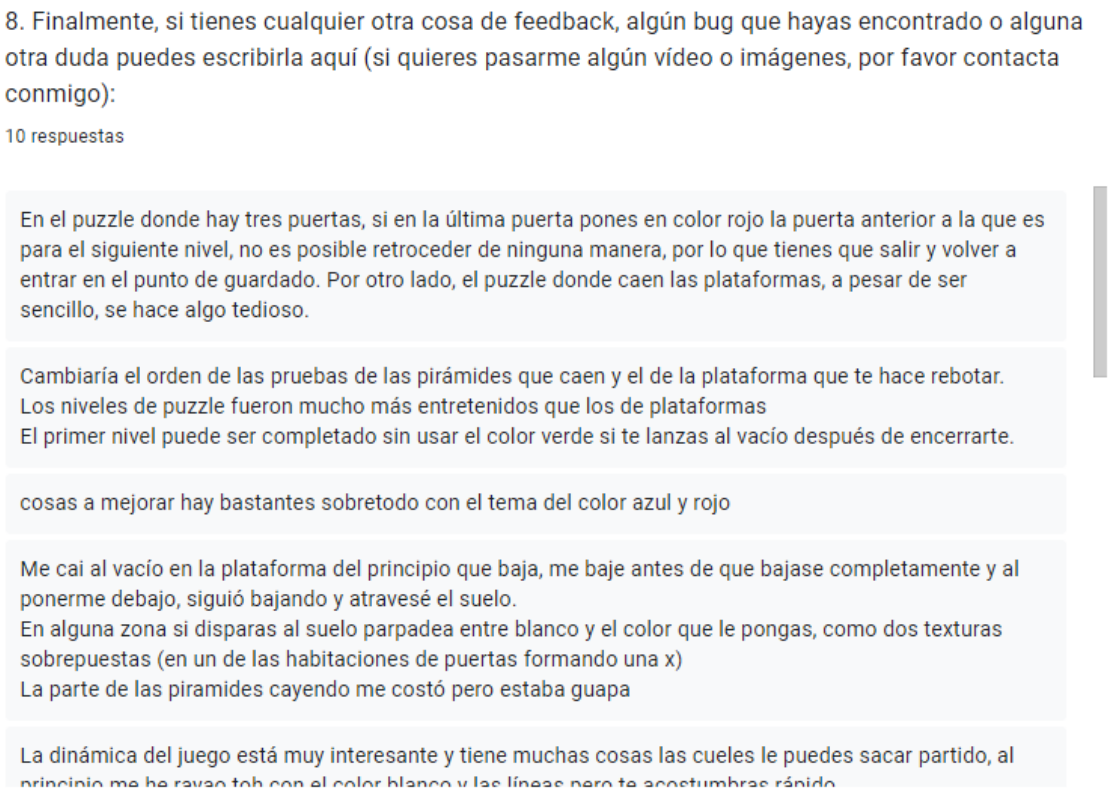


Figura 43. Pregunta de feedback del formulario de testeo. (Fuente: elaboración propia)

4. Finalmente, se ha logrado implementar

Gracias a estos puntos, se ha cumplido la finalidad de este Trabajo de Fin de Grado: Demostrar que es posible guiar al jugador sin tener que recurrir a elementos explícitos como vídeos y fotos.

Capítulo 4. FUTURAS LÍNEAS DE TRABAJO

De cara al futuro, este proyecto se puede mejorar bastante. A continuación, se mostrará una pequeña lista de futuras líneas de trabajo que serían posibles desarrollar:

- Diseñar e implementar más niveles y mecánicas de diferentes colores.
- Agregar un componente narrativo.
- Añadir Easter eggs (o secretos al juego).
- Pulir el apartado audiovisual.
- Publicar el videojuego en Steam u otras plataformas de videojuegos.
- Crear una campaña de marketing asociado al punto anterior.

Como se puede observar, esta lista consistiría en desarrollar y conseguir un videojuego comercial, ya que actualmente el proyecto se acercaría más a una demo jugable y está lejos de ser un videojuego completo. Con todas estas líneas de trabajo, no solo podría tener una fuente de ingresos para futuros trabajos, sino que también serviría como una herramienta clave para el portfolio. Esto se debe porque actualmente en mi portfolio solo hay videojuegos en la plataforma gratuita itchio, y un videojuego publicado en otras plataformas más grandes y populares como es Steam, sería una gran carta de presentación de lo que sería capaz de realizar.

Capítulo 5. CONCLUSIONES

Como conclusión, se puede decir que sí que es posible guiar al jugador sin tener que recurrir a elementos explícitos de aprendizaje como vídeos o pantallas repletas de información sobre controles. Se ha logrado desarrollar un videojuego en el que el jugador simplemente mirando al entorno, logra avanzar de nivel en nivel hasta completarlo.

El concepto de este proyecto había evolucionado bastante desde sus comienzos en mayo de 2023 hasta estos últimos meses. Al comienzo, la idea que se había planteado era centrarse en algún tema relacionado con enfermedades y accesibilidad. En este caso en concreto, se había planteado enfocar el TFG en el daltonismo y desarrollar un videojuego acorde a este tema. Esta era una idea que realmente me llamaba mucho la atención.

Sin embargo, a medida que pasaban los meses de verano, la motivación disminuía. Sentía que este concepto no me aportaría mucho más aparte de ampliar mis conocimientos sobre el daltonismo y accesibilidad.

Entonces en el mismo verano, fue cuando jugué a *The Witness* (Thekla Inc., 2016), el mismo videojuego que posteriormente iba a ser la causa principal de este trabajo. Este título, se caracterizaba por no explicar nada al jugador y eso me fascinó. Fue entonces, que se me ocurrió enfocar la investigación sobre este tema. No solo me interesaba el tema lo suficiente, sino que también me aportaría conocimientos de diseño de cara a mi desarrollo profesional como diseñador de videojuegos.

Y de esto nació la idea de este trabajo. Se decidió al final juntar ambas ideas y desarrollar un videojuego de puzzles relacionado con los colores en el que no se te explica nada.

Por otro lado, respecto al desarrollo, me he dado cuenta de algunas de mis debilidades. La principal es la organización, especialmente cuando trabajo solo. Como el trabajo decidí realizarlo solo, no existía una responsabilidad compartida, sino que todo recaía sobre mí. Debido a esto, me estaba constando conseguir resultados y avanzar con este proyecto. Por lo que, me obligué a fijar unos pequeños sprints de una o dos semanas en las que tenía que enseñarle a Luis, mi tutor, mis avances. Gracias a esto, mi productividad se vio aumentada a gran escala y en pocos meses los resultados iban saliendo por sí solos.

Además, experimenté lo importante que es el feedback de los jugadores a un desarrollador de videojuegos. Muchos de los cambios que hubo en la postproducción eran cosas que nunca se me habían pasado por la cabeza, ya que los desarrolladores de un videojuego suelen “intoxicarse” con el proyecto, es decir, que inconscientemente se centran en hacer que el videojuego funcione. Por ello, enviar el videojuego a personas “limpias” ayuda a sacar ideas y comentarios de cosas a mejorar para que el videojuego no se quede en “correcto” sino que logre llegar a ser entretenido.

En definitiva, este trabajo ha sido un reto final en el que se ponían a prueba todos mis conocimientos que había aprendido hasta ahora. No solo conocimientos técnicos sobre diseño,

programación o arte, sino también, elementos más personales como conocer mis límites y mis debilidades e idear una forma de superarlos y seguir adelante.

Capítulo 6. REFERENCIAS

6.1 Referencias bibliográficas: libros y artículos de investigación

Bonner, M. (2016). *Puzzle About The Island – Multi-Perspective Studies on Knowledge in THE WITNESS.*

Cid, E. (2016). *Portal o la ciencia del videojuego. Héroes de papel (Ed.)*

Dominic Arsenault (2009, octubre). *Video Game Genre, Evolution, and Innovation.*

Hanna R. Rodenbaugh, Heidi L. Lujan, David W. Rodenbaugh, y Stephen E. DiCarlo (2014, junio). *Having fun and accepting challenges are natural instincts: jigsaw puzzles to challenge students and test their abilities while having fun!* https://journals.physiology.org/doi/full/10.1152/advan.00117.2013

Hyrynsalmi, S., Klotins, E., Unterkalmsteiner, M., Gorschek, T., Tripathi, N., Pompermaier, L. B., & Prikladnicki, R. (2018). *What is a minimum viable (video) game? towards a research agenda.* In Challenges and Opportunities in the Digital Era: 17th IFIP WG 6.11 Conference on e-Business, e-Services, and e-Society, I3E 2018, Kuwait City, Kuwait, October 30–November 1, 2018, Proceedings 17 (pp. 217-231). Springer International Publishing.

Jesse Schell (2008, agosto). *The art of game design. CRC Press (Ed.)*

Linehan, C., Bellord, G., Kirman, B., Morford, Z. H., & Roche, B. (2014, October). Learning curves: analysing pace and challenge in four successful puzzle games. In *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play* (pp. 181-190).

Mago, Z. (2019). Easter Eggs in Digital Games as a Form of Textual Transcendence (Case Study).

Martins, R. D. S., Raulino, F. D. C. P., Filgueira, A. M., & Burlamaqui, A. A. R. S. D. S. (2019). *SGDDEdu-Model of short game design document for digital educational games.*

Marçal Mora-Cantallops (2018, agosto). *Transhistorical perspective of the puzzle video game genre.*

Matthew M. White (2014, junio). *Learn to Play: Designing Tutorials for Video Games. A K Peters/CRC Press (Ed.)*

Miguel Sicart (2008, diciembre). *Defining Game Mechanics*.

Naciones Unidas (2015, septiembre). *Objetivo 10: Reducir la desigualdad en y entre los países.* https://www.un.org/sustainabledevelopment/es/inequality/

National Eye Institute (2023, agosto). *Tipos de daltonismo.*

https://www.nei.nih.gov/espanol/aprenda-sobre-la-salud-ocular/enfermedades-y-afecciones-de-los-ojos/daltonismo/tipos-de-daltonismo

Pereira, A. M. M. (2014). *El proceso productivo del videojuego: fases de producción/The production process of the game: production phases. Historia y comunicación social.*

Shuangyuan Cao, Fang Liu (2022, noviembre). *Learning to play: understanding in-game tutorials with a pilot study on implicit tutorials.*

Wolf, M. (2001). *The Medium of the Video Game. University of Texas Press (Ed.)*

6.2 Ludografía

Croteam (2014, diciembre). *Talos Principle.*

FromSoftware (2016, marzo). *Dark Souls III.*

FromSoftware (2022, febrero). *Elden Ring.*

Konami (1986, septiembre). *Castlevania.*

Naughty Dog (2007). *Uncharted.*

Nintendo EPD, Monolith Soft (2023, mayo). *The Legend of Zelda: Tears of the Kingdom.*

Nintendo Research & Development 1, Intelligent Systems (1986, agosto). *Metroid*

PlatinumGames (2009, octubre). *Bayonetta.*

Santa Monica Studio (2005, marzo). *God of War.*

Thekla, Inc. (2016, enero). *The Witness.*

Valve Corporation (2007, octubre). *Portal.*

Valve Corporation (2011, abril). *Portal 2.*

Capítulo 7. ANEXOS

- Enlace al GDD: LINK