

# 2021

IES San Vicente

Gonzalo Arques Jiménez

## SILENCIADOS

Documentación del proyecto final del ciclo superior: "Desarrollo de aplicaciones multiplataformas", realizado por "Gonzalo Arques Jiménez", titulado: "Silenciados". Se trata de un videojuego 3D "Survival horror" realizado en "Unity".

## Índice

Índice	1
Introducción	2
Antecedentes	7
Objetivos	9
Desarrollo y resultados (Metodología, discusiones, problemas...)	11
Conclusión	31
Créditos	32

## Introducción

Bienvenido a la corporación **Zesimov**.



*La vida y su evolución, es nuestra prioridad.*

En este documento serás guiado para conocer el desarrollo del proyecto “**87653L**”, nombre en clave: **SILENCIADOS**.

**Nota:** Antes de empezar, se recuerda que se deben de tener los permisos necesarios (tarjeta de identificación de nivel 9) para poder seguir con la lectura del documento. En caso contrario, estarías faltando la integridad de la compañía, y, por ende, acontecerte problemas.

“**Silenciados**” es un videojuego 3D “Survival horror” creado con el motor gráfico “Unity” en C#. Está basado en los famosos videojuegos de la saga “Resident Evil” y “Silent Hill”, concretamente, en sus versiones más clásicas. Por ende, el videojuego tiene un ambiente terrorífico, cámaras fijas y un control tanque, algo que hoy en día, no se suele ver en productos triple A.



Pronto profundizaremos en ello, pero antes veamos los conceptos base y antecedentes para saber cómo es el juego en profundidad:

## Unity

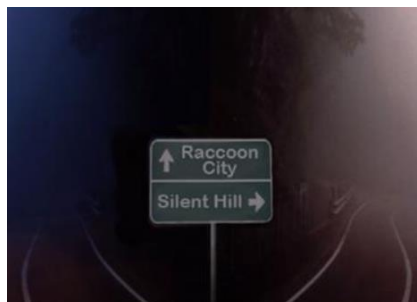
Unity es un motor de videojuego multiplataforma creado por Unity Technologies. Es famoso por ser una muy buena herramienta gratuita (a pesar de tener su plan de pago, llamado “Unity Professional”) para crear videojuegos 2D y 3D, con soporte de compilación en diferentes plataformas (PC, móviles, consolas, etc.).



## Survival horror

Es un subgénero de videojuegos ambientando en el terror/horror, en el que el jugador intenta sobrevivir en ese ámbito. Suelen ubicarse en espacios cerrados, laberínticos y oscuros. Suelen tener su dificultad pues el jugador tiene menos “control” de la situación (a diferencia de los videojuegos de acción típicos), debido a la limitación de munición, salud, velocidad, y visión; o por controles toscos (como podría ser el conocido “control tanque”, del que se profundizará posteriormente), acciones torpes, etc... El jugador debe de resolver puzzles y encontrar elementos que le permitan desbloquear nuevos caminos para continuar con la trama del videojuego. Es decir, se basa mucho en el término “Backtracking” (Volver atrás, en español), concepto jugable que consiste en regresar a zonas del juego que ya se habían visitado, pero que en su momento no eran accesibles, y ahora sí (por haber realizado una acción o haber conseguido un elemento clave).

Aunque tuvo sus previas influencias, el primer videojuego catalogado como “Survival Horror” fue “Resident Evil”, del año 1996.



## Cámaras fijas



A diferencia de otro tipo de cámaras más convencionales como podría ser la cámara en primera persona (Call of Duty, Halo, Minecraft...) o en tercera (Fortnite, GTAV, Warframe...), las cámaras fijas, como bien dice su nombre, son planos fijos en el escenario del videojuego. Es como si el juego estuviese siendo

observado por un guardia de seguridad mirando las cámaras. Esto da gran libertad al director (como si fuera uno de cine) para enfocar el videojuego en planos; y esto, para el terror, es genial. En los Resident Evil clásicos se utilizaba las cámaras fijas para dar tensión al jugador, no se sabía que se iba a encontrar después de una esquina, por lo

que tenía que estar pendiente en todo momento.



El funcionamiento de una cámara fija es simple: Es un plano estático; y cuando el jugador se sale de dicho plano, cambia al siguiente.

Se puede apreciar esto en las dos imágenes anteriores que se muestran junto al texto.

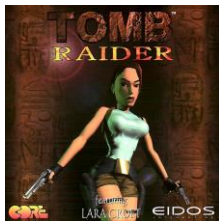
Algunos juegos, prefieren usar otro tipo de cámara, pero utilizar estos planos fijos en ciertas situaciones. Por ejemplo, la cámara que usaba Silent Hill (1999), seguía al personaje, pero, para provocar una mejor ambientación de terror, usaba cámaras fijas en ciertos momentos:



Las cámaras fijas están vinculadas al único tipo de control apto para ellas (algo que he descubierto tras desarrollar este proyecto), el control tanque:

## El control tanque

Es la mayor deficiencia al usar las cámaras fijas. Incluso en su época, era tosco. Por ejemplo, muchos fans de Resident Evil evitan jugar a los clásicos por este tipo de control. Su funcionamiento es que el jugador debe de rotar el personaje, y luego avanzar o retroceder. Como hace mención a su nombre, es similar al movimiento de un tanque de guerra. Es algo difícil de controlar, sobre todo al principio. Lo más típico es que el jugador inexperto esté chocándose constantemente con las paredes por no rotar el personaje a tiempo. Pero es que incluso con jugadores adaptados, a veces puede darles malas pasadas. Este tipo de control se utilizó mucho en la época de Sony Playstation, con juegos como Tomb Raider (a pesar de que este título no usa cámaras fijas).



**Enlace 1.** Gameplay de Tomb Raider (1996), juego que utilizaba el control tanque:

<https://youtu.be/GpEEBU9xcEo?t=457>



**Enlace. 2.** Gif de Resident Evil 2 (1998), juego que utilizaba las cámaras fijas y el control tanque:

<https://static.vix.com/es/sites/default/files/r/resident-evil-2-ambiente.gif>

El juego más reciente (famoso) con (semi) cámaras fijas y control tanque (un poco mejorado), es The Medium:



<https://youtu.be/MmfY4GtBiiM?t=6594>

Bueno, habiendo aclarado la base del videojuego, podemos empezar con los demás apartados.

## Antecedentes

Desde que conocí la saga de Resident Evil, he amado el subgénero “Survival horror”. Cuando completé muchos de sus títulos, busqué algunos similares, y me encontré con sagas como “Silent Hill”, “Dino Crisis”, “Fatal Frame” y muchas más. Juegos más modernos podrían ser “Outlast” (del cual, también saco inspiración para este proyecto), “The Evil Within”, “Dead Space”...

Y siempre he querido crear mi propio “Survival horror”, por lo que, este proyecto es un acercamiento a ese deseo. Veamos un vistazo rápido a mis inspiraciones principales (centrándome en sus primeros títulos), adjuntando un enlace para ver un gameplay:

### Resident Evil



Sin duda alguna, es mi obsesión. Amo su jugabilidad (sobre todo la clásica de la que hablaremos ahora), su historia, sus personajes, su universo en sí. Desde el principio con Resident Evil (1996), ubicado en la terrorífica mansión del fundador de la malvada corporación Umbrella, hasta el más reciente título: Resident Evil Village. Una saga de videojuegos que ha tenido un fuerte impacto en mi vida.

Debido a las limitaciones de la videoconsola (Sony Playstation) en la que surgió el primer videojuego, se ideó como un juego de terror con cámaras fijas y control tanque. El jugador se encontraba acorralado en un escenario repleto de armas biológicas (zombis y otro tipo de monstruos) que lo acechan violentamente, sumando una munición y componentes curativos extremadamente escasos para defenderse. Además de encontrarse con puzzles a resolver, puertas cerradas que había que desbloquear (encontrando las llaves correspondientes) y demás retos. Los desarrolladores no sabían que género asignarle, por lo que crearon el término: “Survival horror”. El juego tiene un tono de serie B, por lo que su historia es sencilla, aunque intrigante. Cuenta la mayoría de los sucesos a través de documentos escritos que el jugador puede leer opcionalmente. Esto es algo muy habitual de los “Survival horror”.

<https://youtu.be/6jBqGn41HPI?t=1134>



### **Silent Hill**



Es una de las principales sagas de “Survival horror”. Es obvio que tomó inspiración de Resident Evil, pues comparte su jugabilidad y forma de contar la historia. Sin embargo, tiene su propia esencia y un terror distinto: Mucho más psicológico, japonés, sobrenatural, y personal.

[https://youtu.be/tny6nZHf\\_08?t=275](https://youtu.be/tny6nZHf_08?t=275)

### **Outlast**



Es un juego más reciente, por lo que también, el más diferente. No usa cámaras fijas, ni control tanque. Sin embargo, mantiene elementos típicos de los “Survival horror”, como el contar la historia a través de documentos, por ejemplo. Su terror se basa en la tensión de ser perseguido por los psicópatas de un viejo asilo.

<https://youtu.be/-1d2HGJCCd0?t=3451>

El videojuego de este documento (Silenciados), reúne elementos de estos tres maestros, y en los próximos apartados, podrás identificarlos. Incluso hay referencias.

## Objetivos

La idea era crear un videojuego en Unity 3D, orientado al subgénero “Survival horror”, inspirado en los famosos videojuegos de Silent Hill, y Resident Evil, concretamente, los clásicos (Resident Evil 1, 2, 3, Code Veronica X...). Es decir, un juego de cámaras fijas en el que el protagonista se encontraba atrapado en un escenario cerrado (una mansión, un laboratorio, o algo por el estilo), y repleto cosas por hacer: objetos, documentos y puzzles. Debía de moverse por el entorno buscando objetos que le permitan completar puzzles para poder avanzar en la partida (Llaves que abren puertas, objetos útiles para otros puzzles, etc.). El problema es que el escenario estaría repleto de enemigos (o de uno solo, pero muy poderoso, que lo perseguiría), y el protagonista no podría acabar con todos por la escasez de recursos defensivos, y por sus pésimas habilidades de combate (es un simple (pero galardonado) periodista independiente), así que, tendría que arreglárselas para no ser asesinado mientras intenta escapar del lugar resolviendo los puzzles y yendo de un lado a otro. La ambientación (oscuridad, diseños, música, sonidos...) estaría muy centrada en el terror y la tensión, poniendo mucho empeño en ello.

Los objetivos generales a realizar eran estos:

- Crear un escenario grande y relativamente laberíntico repleto de salas, ambientadas en el terror (y con la idea previamente diseñada de cómo se implementarán los puzzles en el escenario).
- Crear un correcto movimiento de jugador junto a las cámaras fijas (colocadas por el escenario).
- Crear un sistema de interacción de elementos con el jugador. Por ejemplo, si inspecciona una puerta cerrada, salga un texto similar a: “La puerta está cerrada. Tiene un símbolo de trébol en su hendidura.”, indicando que no se puede acceder si no se tiene ese “algo” con símbolo de trébol. O si simplemente se interactúa con un reloj: “Es un reloj antiguo y roto, marca las 12:30”.
- Crear un sistema de inventario para poder almacenar los objetos recogidos (con la idea de que sea limitado). Y con opciones similares a: “Usar”, “Comprobar”, “Combinar”. Y en caso de tener un inventario limitado, crear un sistema de cofres (conectados entre sí) para poder almacenar los objetos sobrantes del inventario.
- Crear un sistema de lectura de documentos, y repartirlos por el escenario para ampliar la historia del juego (por ejemplo, un diario de una persona antes de convertirse en un enemigo al que hay que enfrentar) o dar pistas sobre cómo resolver los puzzles.

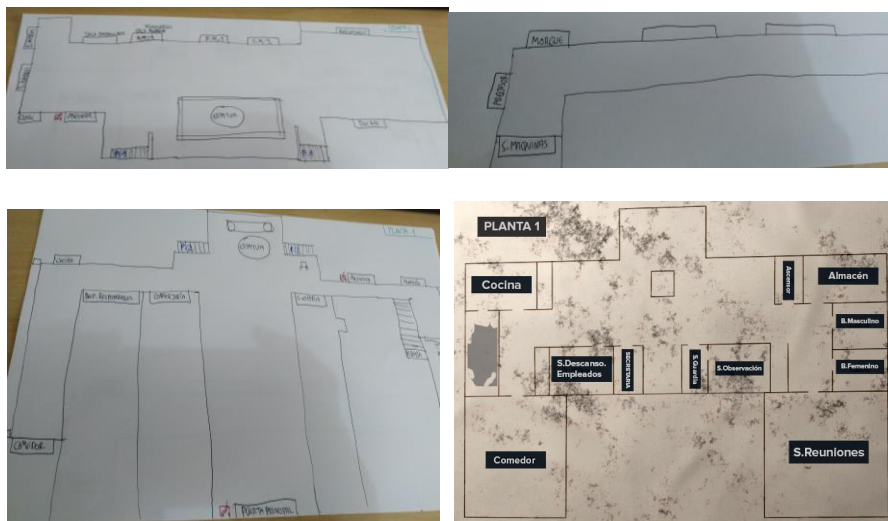
- Crear a los enemigos y su comportamiento (o un único enemigo poderoso) para repartirlos por el mapa.
- Crear un sistema de defensa interesante entre el jugador y los enemigos.
- Crear los objetos recogibles y repartirlos por el escenario.
- Crear los puzles en el escenario (que se relacionen con los objetos recogibles, como podría ser una llave que abra una puerta).

## Desarrollo y resultados (Metodología, discusiones, problemas...)

### Preliminares y diseño del nivel principal

Lo primero que decidí hacer fue buscar assets en la tienda de Unity (y en otras webs). Para ver qué cosas podía hacer y qué no. Tras un largo tiempo recopilando assets gratuitos, junto a uno de pago (unas llaves en forma de piezas de ajedrez) que adquirí, elegí un escenario gratuito de un viejo asilo terrorífico. Lo usaría como escenario principal de mi juego y lo adaptaría a mis otros assets e historia pensada. También uso uno de un laboratorio, un tablero de ajedrez, y uno construido por mí a base de assets.

Así que, teniendo el escenario principal del juego listo, junto al resto de assets, empecé a diseñar el nivel del juego con esos elementos. No en cuanto a la programación, sino, a todos los pasos que hay que realizar para completar el videojuego: Los objetos que debía colocar en cada lugar, los puzles que iba a incluir, las puertas (y su tipo de cerradura) que hay que abrir, y demás elementos... Por lo que dibujé el mapa del escenario (y todas sus plantas) en papel para tener una vista “desde el cielo” de las numerosas habitaciones que tiene (luego lo diseñé mejor en “Photoshop” para incluirlo como asset dentro del videojuego).



Entonces, ayudándome con el mapa, empecé a construir el “laberíntico” nivel en un documento escrito, escribiendo todos los pasos a realizar para completar el escenario.

A continuación, dejo una guía de como completar el videojuego (la parte del asilo), que es básicamente lo que escribí en su momento, pero modificado y mejorado:

## Diseño del nivel del asilo (Guía de juego)

El objetivo principal del escenario del asilo, es insertar tres estatuas en forma de grifos (seres mitológicos) en la estatua del Main hall.

**Nota:** Todas las capturas que acompañan al texto son tomadas de la versión “final”.

### GRIFO DE MADERA

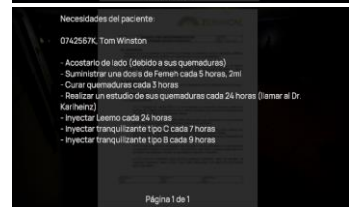
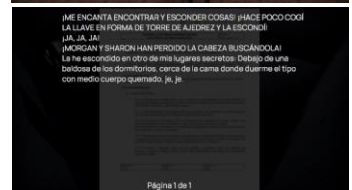


Ir a **secretaria** (dónde se puede ver el **mapa** y una **caja de seguridad** que requiere código (-[img alt="QR code" data-bbox="595 355 615 375])).



Ir a **cocina** y **coger la bombilla** [img alt="Light bulb icon" data-bbox="438 395 458 415]] de una lampara a medio reparar.

Ir a **sala de reuniones** y **poner la bombilla en el proyector** para que se encienda y **muestre el código de la caja de seguridad de secretaria** [img alt="QR code" data-bbox="515 495 535 515]].



Volver a **secretaria** y **poner el código para recibir la llave peón** [img alt="Pawn icon" data-bbox="285 535 305 555]].

Ir a la **sala de descanso de empleados** (img alt="Pawn icon" data-bbox="510 555 530 575]), **utilizar la llave de peón en ella (para usarla y que se descarte antes)**.

Ir a la **sala de dormitorios** (img alt="Pawn icon" data-bbox="435 590 455 610]) **usando la llave de peón, y resolver el puzle de la baldosa** (pulsar INTERACCION en la baldosa correspondiente con la **ayuda del documento que está en la ducha**) para **obtener la llave torre** [img alt="Tower key icon" data-bbox="520 635 540 655]].



Ir al **almacén** (img alt="Tower key icon" data-bbox="540 650 560 670]), **utilizar llave torre, coger la ganzúa** [img alt="Pick icon" data-bbox="605 665 625 685]] y **un jarrón** [img alt="Vase icon" data-bbox="525 685 545 705]].

Ir dónde está el **extintor** (img alt="Extinguisher icon" data-bbox="615 700 635 720]) y **usar la ganzúa** (posible puzle) para **cogerlo** [img alt="Extinguisher icon" data-bbox="545 735 565 755]].





Ir a la **cocina** y **usar el extintor en el fuego**.  
Coger grifo de **madera** [🔧].



## GRIFO DE MÁRMOL



Ir al **comedor** y coger la llave caballo [♞].

Ir a **sala médica B** (♟), **utilizar llave peón**. Hay un **contenedor para crear un disolvente**. Se deben de **juntar tres componentes en este contenedor para crearlo**: Uno de los componentes

(Acetona) está detrás de la estatua del Main hall [♚]. Otro

(Vinagre) en el congelador de una nevera de la cocina [♚]. Otro

(Éter etílico) en la sala de descanso de empleados [♚].

Teniendo el disolvente listo y recogido [🧪], ir a la **morgue** y **disolver**

la **silicona que atrapa a la llave alfil** [♜], y entonces, la podemos obtener.



Ir a la **cocina de los empleados** (♞) y **usar la llave alfil**. Coger llave inglesa [🔧].





Ir a la **sala de máquinas** (♞), **usar la llave caballo**, y **arreglar motor hidráulico con la llave inglesa**. Esto hace que las llaves de los grifos de los lavabos funcionen y den agua.





Ir a una llave de grifo y llenar el jarrón con agua.





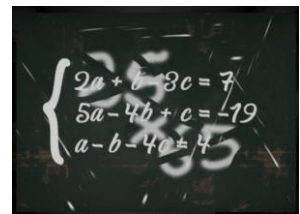


Ir a **habitación de reclusión A** () , usar **llave caballo**, y **utilizar el jarrón con agua en una pared que tiene una mancha negra** para mostrar el código [





Ir a la **sala guardia -> sala de observación** () , **utilizar llave alfil en las puertas**, y **poner el código en la caja de seguridad**. **Coger llave del rey** [] de la caja.

Ir a la **sala de educación** () y **usar la llave de la torre**. **Resolver puzle de la pizarra** para **obtener un código** [





complicada, sin embargo, solo hay que fijarse bien en una multiplicación sencilla que está sobre la operación complicada.

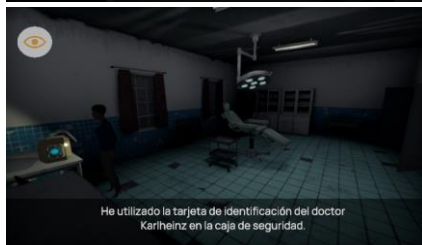



Ir al **despacho del médico jefe** () y **usar la llave del rey**, **utilizar el código de la sala de educación en una caja de seguridad** para **obtener el grifo de mármol** [].

## GRIFO DE BRONCE



Ir a la **sala médica A** () y **utilizar la llave caballo**. Hay una caja de seguridad que requiere una tarjeta de identificación para poder abrirse (-  -). **Se necesita coger la tarjeta correcta**. Hay tres tarjetas: Una en el almacén, otro en la sala médica B, y la última (y correcta) en la



**habitación de reclusión B** () (que al acercarse aparece un “jumpscare”). **La habitación de reclusión B se abre con la llave alfil**.

**Cuando se use la tarjeta de identificación correcta en la caja, obtenemos un tubo curvo con válvula**.





Vamos a la **sala de máquinas (2)**, y usamos el tubo con válvula y la llave inglesa para eliminar el vapor que bloqueaba el

paso, y entonces, ya podemos acceder a una nueva caja de seguridad con código (-). Para saber la contraseña hay que ir a la **cocina de**

**los empleados (♟)**, y leer una nota que indica que la contraseña de la caja es la fecha de cumpleaños del responsable. Y si vamos a la **sala de empleados (♟)**, hay una nota que indica que cuando fue el cumpleaños de ese empleado. Por lo que, al **introducir la contraseña en la caja de la sala de máquinas**, obtenemos el grifo de bronce [♟].

Hace unos días, el equipo del laboratorio trajo unas cuantas cajas de seguridad. Alfred se encargó de una, y la guardó en la sala de máquinas (que es su zona de trabajo). Como está desaparecido y es posible que la señora Wilkes quiera acceder a ella en algún momento espontáneo, Alfred nos dijo que la contraseña para abrirla era su día y mes de nacimiento (en formato: ddmm), cosa que ya deberíamos saber...

Así que, si Wilkes os pregunta por dicha caja, ya sabéis como responder.

Morgan Sanderson

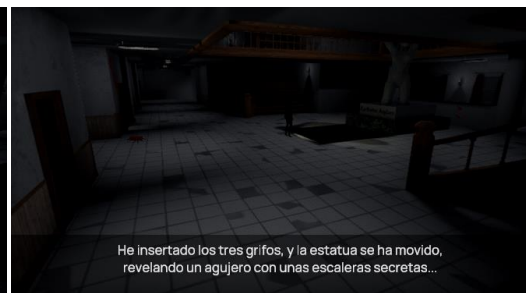
Página 1 de 1

13, Junio

[Hoy hemos dado una sorpresa a Alfred! ¡Cumple 49 años! Es un gran trabajador y aún más importante, una excelente persona. Le hemos regalado entre todos un reloj dorado de Maren. Le ha encantado, es un fanático de esa marca. Por gente como él, sigo aquí. Alfred, Morgan, Sasha, y el resto, son buenos compañeros... El problema es la señorita Wilkes y el doctor Karlheinz... Cuando alguno de los dos aparece por aquí, el buen rollo se esfuma, y el ambiente se vuelve frío y negativo...

Página 2 de 5

Con los tres grifos obtenidos, se utilizan sobre la estatua del Main hall y entonces, se abrirá un hueco secreto debajo de ella, llevando a la protagonista a un nuevo ambiente...





Esto es todo lo que hay que hacer en el escenario del asilo (el juego principal) para pasar a la siguiente parte del videojuego (de la que informaré luego). Me hubiera gustado incluir más cosas, y mejorar las que ya están (como el inventario), pero ya hablaré de eso más adelante. La verdad, es que estoy realmente contento con el resultado.

Como se puede apreciar, hay bastante por hacer, y mucho “Backtracking”. Esta parte del asilo, tiene una duración media de una hora de juego (incluyendo la lectura de documentos de la que aún no he hablado). La mayoría de las pruebas con los beta tester obtenían este tiempo. Por lo que sumando el resto de escenarios y cinemáticas, el juego tiene una duración de más de una hora.

No hay elementos repetitivos. Es decir, por cada “objetivo” a hacer en el juego, tenía que diseñar algo nuevo: En uno añadir un fuego, en otro, hacer que una baldosa se levante, en otro encender un proyector... No había nada que me permitiera simplemente copiar y pegar.

Programé las cosas en el orden en que lo escribí, sin embargo, no es el único orden para completarlo. Hay muchas formas alternativas, y, de hecho, ninguno de mis beta testers lo jugaron en el orden en el que lo planteé. Incluso recogían al final el grifo de madera, que es el primero que diseñé.

Como ya he adelantado, también he creado una trama para el videojuego. Me encanta escribir historias y no podía evitarlo. El juego está repleto de documentos que explican la trama del título. También sirven para resolver puzzles o explicar elementos del juego. No es coincidencia que el villano esté obsesionado con la musa Urania, el Dios Apolo y sus grifos, y luego, en el juego, haya estatuas de ellos... En mi cabeza hay mucha más historia de la que hay plasmada por lo que me hubiera gustado incluir más documentos y “cinemáticas”. Incluso tenía pensando en esconder algunos opcionales en cajones que debían ser desbloqueados con la ganzúa.

Resumen de la historia planteada: Rose es una galardonada periodista independiente. Su hermana, Sharon, trabaja como enfermera en el aislado asilo Karlheinz, propiedad de la multinacional: “Zesimov”. Sharon le alertó a Rose de que en ese asilo se estaba cociendo algo malo, y cuando una noche, ella no apareció, Rose fue en su búsqueda. Cuando la periodista llegó al viejo edificio, se encontró con un lugar vacío, lleno de muerte y misterio. Repleto de documentos de las personas que allí moraban: Enfermeros, celadores, médicos, etc... Pronto, descubrirá que su hermana tenía razón, Zesimov estaba experimentando con los pacientes, teniendo un laboratorio bajo el asilo. Algo salió mal ahí abajo, y provocó un caos en todo el edificio...

Anton es el paciente más problemático que tenemos actualmente. Su triste condición, sumada a su terrible actitud y educación, le hacen una persona terriblemente conflictiva.

Es deleznable que manoseé a las enfermeras (sobre todo a Sasha) y les robe sus pertenencias para luego esconderlas. Y no solo coge los bienes de nuestras amadas enfermeras, sino también elementos indispensables del asilo, como podría ser una llave de alta seguridad...

Los tranquilizantes pueden calmarlo temporalmente, pero no son suficientes para modular su conducta. Así que, a modo de castigo correctivo, sugiero que el señor Oliveira pase un tiempo en la habitación de reclusión B. Mientras esté ahí, sólo se le darán los cuidados básicos, y no se le permitirá bajo ningún concepto salir de la habitación hasta el tiempo estipulado por el que decida el doctor Karlheinz.

Espero que esto ayude a corregir su conducta. Cualquier consulta, háganmela saber.

Enfermera A Wilkes

< Página 2 de 2

Resultados desastrosos. Ninguna mujer ha sido apta, solo dos hombres (los más jóvenes). Aunque hemos obtenido un resultado interesante: Un hombre ha mutado. Por lo que investigaremos sus alteraciones genéticas antes de eliminarlo. Solicito su permiso para ello.

P.D: Debido a estos resultados, en las futuras pruebas deberíamos de usar un rango de edad menor...

< Página 4 de 4

Me he centrado en crear personajes interesantes: La malvada enfermera jefe A.Wilkes (cuyo nombre es una referencia a la villana del libro de Stephen King: Misery), el doctor Karlheinz, Anton Oliveira (un loco que le encanta robar cosas y esconderlas por ahí, lo que explica que muchos objetos estén repartidos), etc.

Apolo: Dios de la muerte súbita, de las plagas, y de las enfermedades... pero también, el dios de la curación, y de la protección contra las fuerzas malignas.  
Patrón de la salud, la belleza, la música, y las Bellas Artes.  
Presidía las leyes de la religión, y las constituciones de las ciudades.

Hacía a las personas conscientes de sus pecados, y era el agente de su purificación...

Era temido por los otros dioses... Su padre era el único que podía contenerlo.

Apolo siempre iba cabalgando con sus grifos por los cielos. Y cada día, visitaba a una de sus hermosas musas.  
Su favorita: Urania, musa de la Astronomía y la Astrología.

Página 1 de 2 >

## Manos a la obra

Bueno, teniendo el diseño del nivel y la historia planteados y adaptados a los medios de los que disponía, empecé con el “trabajo” de verdad. Abrí Unity, y comencé por probar el movimiento del jugador en el escenario. Y es aquí, dónde me di cuenta de que ninguno de los controles que encontraba ya hechos por la Unity Asset Store, me valía para las cámaras fijas (aún no estaba desarrollado, pero tenía unas de pruebas) que quería utilizar. No eran aptos para este sistema de cámaras, ya que, eran muy incontrolables. Entonces entendí porque en los juegos en la que las utilizan, se decantan por el control tanque (incluso los modernos, como The Medium). Por lo que, me puse a programarlo por mi cuenta. Al fin y al cabo, es posiblemente, el control más sencillo de programar (lo hicimos en este mismo curso).

Cuando terminé con el control, descargué un asset y animaciones gratuitas (de la página de Adobe: Mixamo) para el personaje que controlaría el jugador. Lo incluí y diseñé para que, según el movimiento, aplicará una animación u otra. Además, tuve que modificar las animaciones para incluir un sonido de pisadas en cada paso que hacía el personaje. Pudiendo mover al personaje (con su aspecto y animaciones) por el escenario, le implanté una cámara en tercera persona temporal para recorrerlo mejor.

Lo siguiente que desarrollé, fue el tema de abrir y cerrar puertas. Es relativamente más complejo de lo que parece. Descubrí otra cosa de la que no te percatas a la hora de jugar videojuegos. En el mío, cuando se cierra o se abre una puerta, si el jugador está muy cerca, le golpea, cosa antinatural desde luego. Y es que, en los juegos (3D), se suele tener una animación para abrir puertas (cosa que yo no podía hacer), o funcionan de forma antinatural también (pero mejor que lo mío). Por ejemplo, en los últimos Resident Evil, el jugador puede atravesarlas rápidamente sin que el personaje haga una animación, pero éstas no se mueven como lo hacen en la vida real, las cuales solo tiene un movimiento (generalmente). En estos juegos, sin importar si el personaje entra o sale, la puerta se desplaza al lado correspondiente. Básicamente cómo funcionan las típicas puertas de los bares del viejo oeste.



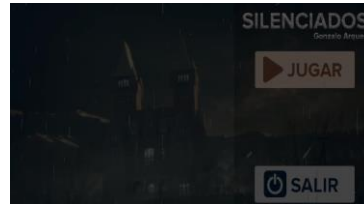
**Nota:** Como se aprecia en las imágenes, abriendo la misma puerta en ambos lados, la puerta se desplaza igual, cosa relativamente antinatural para este tipo de puertas, pero funcional.

Después de las puertas, programé más cosas del juego base: El sistema de recogida de objetos, inventario y su animación (que solo es un visualizador de los objetos que se tienen actualmente), lector de documentos sencillo (posteriormente lo mejoré), interacción con elementos, una interfaz para mostrar un mensaje informativo (al inspeccionar un elemento, por ejemplo), un indicador de interacción que se colocaba cerca del elemento a interactuar (esto cambió posteriormente), etc.).



Cuando tuve toda la base hecha, comencé a realizar todo lo que hay que hacer para conseguir la primera estatua del grifo. No fue especialmente complicado a excepción del sistema de cajas de seguridad con código. Quizás si la interfaz hubiese sido con el ratón, sería más fácil porque cada número es un botón y ya. Sin embargo, me obcequé en que el

control fuera totalmente por teclado. No quería que el ratón se tocara para nada en mi videojuego (cosa que al final no fue así, ya hablaremos de ello, cuando llegué el combate). Aunque fue costoso, lo logré, y adapté este control para realizar también, el menú principal y sus opciones. El fondo del menú principal es un vídeo que realicé en un programa de edición (Camtasia Studio 9), y que averigüé como insertarlo en Unity, haciendo que se vea bien sin importar el tamaño de la pantalla (como todo el juego, en realidad). Todas las “cinemáticas” del juego las hice así.

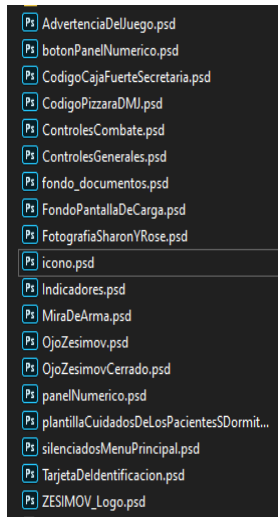


Volviendo al tema de la caja, se permite un código máximo de cuatro dígitos, y si se ha introducido uno incorrecto o correcto, muestra una animación correspondiente (el código se ilumina en verde o en rojo).

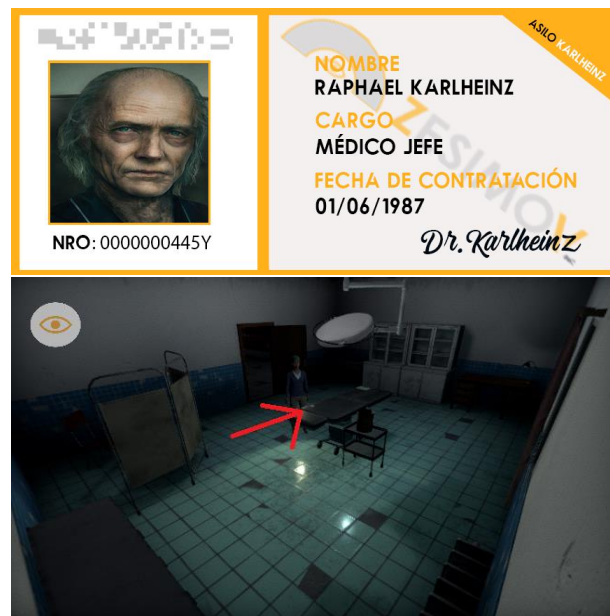
En este punto que ya tenía la base del juego hecha, y los pasos realizados para conseguir el primer grifo, decidí adaptar el escenario a lo que tenía planificado (también en cuanto a la historia). Lo modifiqué, le añadí y quité assets, arreglé algunos problemas que tenía de base, implementé un sistema de iluminación para una mejor ambientación, mejoré los gráficos, añadí música (que, dependiendo la zona, cambia) y efectos de sonido para ambientar más al terror (lluvia, ruidos extraños...), etc. También decidí añadir todos los objetos del juego y sus dependencias (partículas, sonidos (algunos con audio 3D), etc.).

Esto me llevó su tiempo, pero creo que no tanto como el de añadir bloques invisibles por todo el escenario. No sé exactamente el motivo (no sé si es el escenario, mi control de jugador o qué), pero el jugador podía atravesar las paredes y puertas con facilidad. Esto es muy grave, no solo porque puede “caerse” del escenario y ya no regresar, sino

que también rompe por completo mi diseño de “Survival horror”. Si el jugador puede atravesar las cosas, ¿para qué va a calentarse la cabeza buscando tal llave para tal puerta? Así que, mi solución fue colocar estas “paredes” invisibles que bloqueaban el paso. Estuve mucho tiempo probando, y creo que el jugador ya no puede atravesar nada.



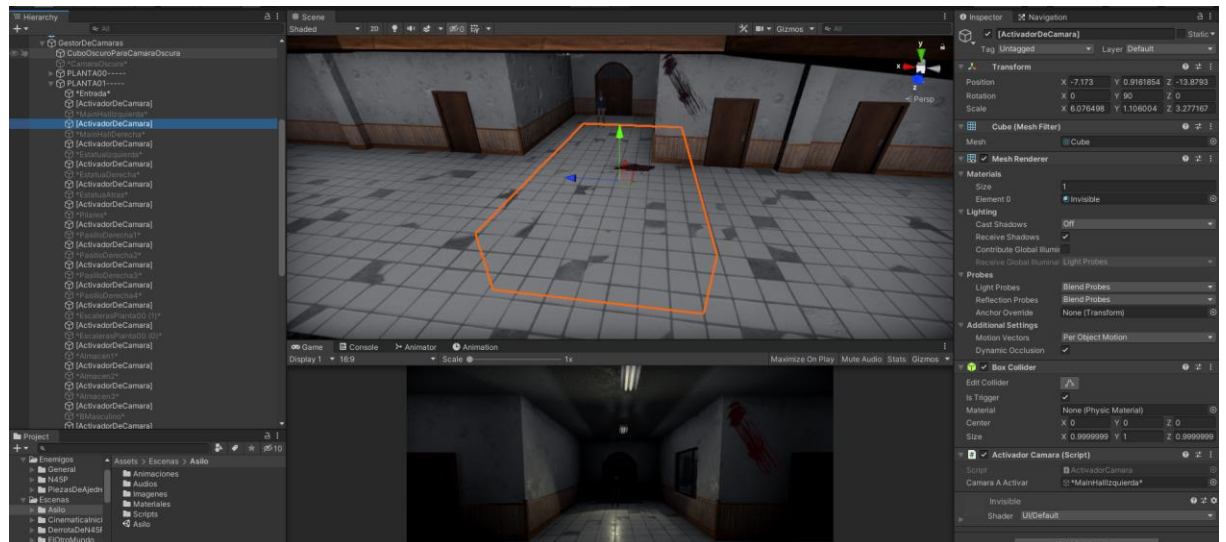
En la adaptación del escenario, también hay muchas imágenes creadas por mí (con Photoshop) para la decoración, interfaces, mapas de las tres plantas del asilo, etc. De hecho, considero que he sido detallista, porque he diseñado cosas que en el juego ni se llegan a apreciar, como podrían ser las tarjetas de identificación de los empleados del asilo:



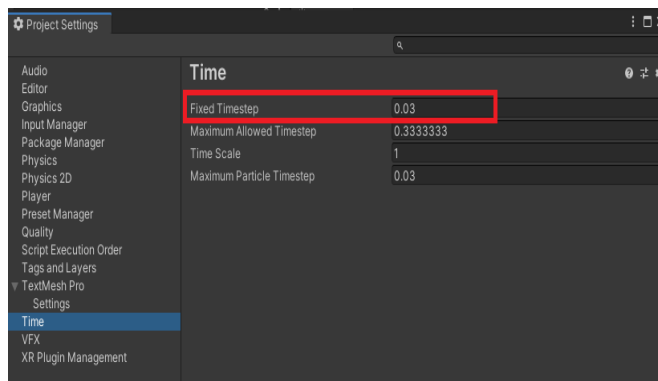
Después de preparar el escenario, continué con el resto de pasos a realizar para completar el juego, es decir, conseguir los dos grifos restantes y colocarlos en la estatua del Main hall. No tuve muchas dificultades. Y como ya tenía todo bien planificado, solo tenía que seguir el orden que planteé.

Cuando ya tuve la base del juego y el nivel del asilo “completados”, implementé las cámaras fijas, y aquí sí que tuve “problemas”. En primer lugar, intenté usar el famoso “Cinemachine” de la Unity Asset Store, pues ya tiene un sistema integrado para simular las cámaras fijas, sin embargo, no funcionaba exactamente como yo quería. Así que, me ocurrió algo similar a los controles: “Si quieres algo bien hecho, hazlo tú”.

La idea parecía sencilla: Poner unos “activadores” por el escenario asociados a unas cámaras. Y realmente así está hecho, cada activador tiene asociado una cámara. Cuando el jugador pasa por ese activador (que cubre toda la zona que enfoca la cámara), desactiva todas las cámaras (aunque realmente, por temas de rendimiento, ya lo estaban, a excepción de la que estaba activa en ese momento), y entonces activa la nueva.



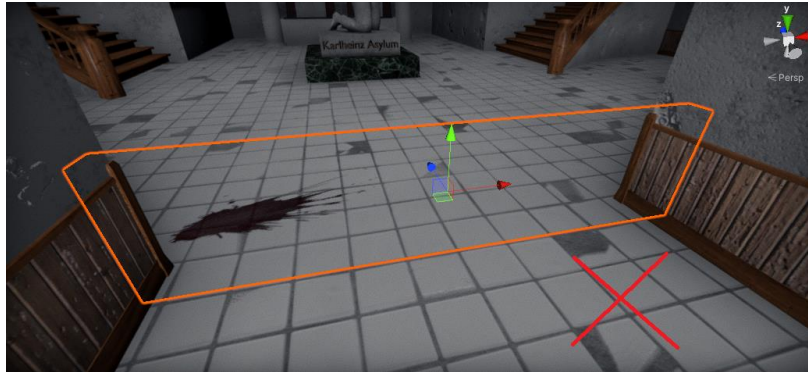
El problema es que a veces, el jugador pasaba a un nuevo activador, y la cámara no cambiaba, se quedaba en la anterior, haciendo que el jugador no viese al personaje. Esto



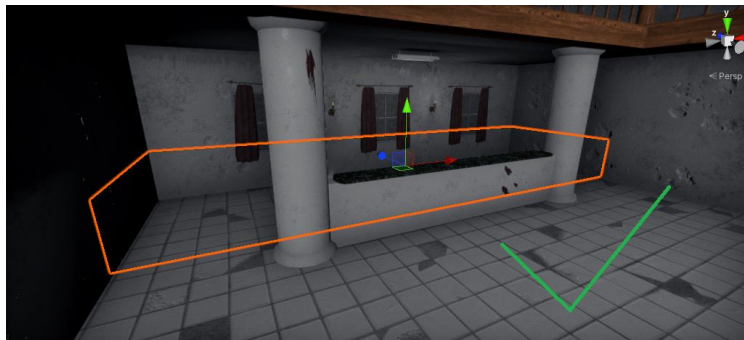
me volvió loco. Estuve probando mil cosas y al final encontré la solución. No era algo mío, sino una configuración del tiempo de Unity en los parámetros del proyecto.

Este problema fue un infierno para mí...

Otra cosa a tener en cuenta, es que como he dicho, el activador debe cubrir todo el enfoque de la cámara, ya que si lo hiciésemos como si fuera una especie de barrera automática para el parking de los coches, el jugador vería la cámara anterior hasta pasar por esa barrera, lo que conllevaría a que el jugador no tuviera visión del personaje.



**Fig. 1.** Una barrera. Incorrecto en la mayoría de los casos.



**Fig. 2.** Cubre toda la zona de esa cámara fija. Lo correcto.



Al final han sido unas 69 cámaras (y unos cuantos activadores más) en el asilo, y otras más en el resto de escenarios, superando las 100 cámaras fijas en todo el juego.



Colocar este tipo de cámaras, me ha hecho sentirme como si fuera un director de cine. Siempre y cuando no afectase a la jugabilidad, las colocaba artísticamente a mi gusto, o enfocando algo en concreto.



A pesar de que pueden dar algún problema de jugabilidad y la obligación de poner el control tanque, me encanta este tipo de cámara.



Eso sí, al introducirla, tuve que cambiar unos detalles. Por ejemplo, desarrollé un indicador de interacción que se colocaba cerca del elemento a interactuar, pero al introducir las cámaras fijas no funcionaba como debía, por lo que, lo cambié para que simplemente apareciera en una esquina de la pantalla:



Y uno de los problemas de jugabilidad que puede dar, es el combate. Añadir armas en un juego en tercera persona cuando no se tienen animaciones, es complicado, y en este caso, ocurre lo mismo, e incluso peor. Y claro, esto va de la mano con el tema del enemigo.



Hombres:

C.García - Apto
A.Oliveira - Apto
L. Hidalgo - Fallecido
B. Robinson - Fallecido
M. Coen - Fallecido
M. Smith - Fallecido
T. Winston - Mutado

La implementación del enemigo es lo que más me costó planificar. Tenía claro que el principal iba a ser una luz parpadeante. Puede que sea sencillo, pero me gustó la idea. Sería el responsable de todo lo que sucedió en el asilo (aunque realmente fue culpa de la empresa malvada). También tenía pensado implementar dos enemigos más, una especie de zombis mecánicos, pero fueron descartados.

El escenario del asilo era muy cerrado, y con muchas habitaciones con puertas. De hecho, el personaje jugable ya se encuentra estrecho en la mayoría de las habitaciones. Incluir enemigos aquí sería desastroso. Así que me planteé varios casos para la creación de los enemigos y la defensa del jugador contra él. Las más razonables fueron que fuese un único (relativamente) enemigo, que, al alcanzar al jugador, lo llevase a otro mundo, un mundo de pesadilla, alejado del escenario del asilo, y entonces ahí, el jugador podría realizar la defensa contra el mismo. Se me ocurrieron varias ideas respecto a esto, pero solo probé dos y ambas fueron relativamente descartas:

- Una persecución en un tablero de ajedrez infinito. El jugador correría por un tablero de ajedrez sin fin, mientras el enemigo le perseguía. Las piezas de ajedrez del tablero, se moverían aleatoriamente hasta el jugador (realizando el movimiento correspondiente de la pieza, es decir, el caballo haría su movimiento en L), bloqueándole el paso, para ayudar a que el enemigo lo alcanzase.

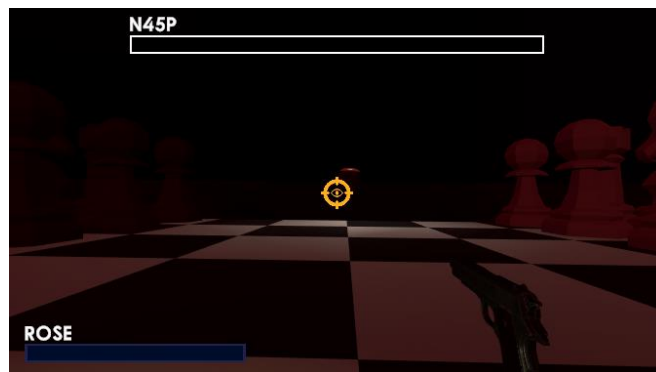
Esto fue descartado por muchos motivos (movimiento de las piezas (relacionadas entre sí), escenario infinito, animaciones de las piezas “vinculadas” al escenario infinito), pero el principal es que requería mucho tiempo de trabajo, el cual ya era muy escaso...

- Un combate en un tablero de ajedrez enorme. El enemigo estaría en una parte del escenario. El jugador tendría un arma, y podía moverse por el tablero libremente. Su objetivo sería el de eliminar al enemigo, mientras derrotaba a las piezas de ajedrez que su adversario invocaba para detenerle. Esta vez, las piezas no se moverían según su movimiento correspondiente, sino que irían directamente hacia el jugador.

Esto tenía más posibilidades, incluso implementé nuevas animaciones de movimiento al personaje para que se sintiera más frenético, pero fue descartado principalmente por dos motivos: El primero es que el enemigo es una luz que parpadea, y al ser un escenario tan gigantesco, la luz solo rebotaba contra el suelo, y quedaba cutre. Y el segundo motivo, es que la cámara en esta ocasión, estaba sobre el hombro del personaje, es decir, en tercera persona. Y como he mencionado, al no tener animaciones, cuadrar el arma con la animación del personaje es algo muy difícil. Realicé unos scripts para intentar conseguirlo, y aunque logré algo interesante, no me convencía del todo. Me hubiera gustado haber puesto alguna imagen de este primer acercamiento al combate contra el enemigo, pero no guardé ninguna en su momento. Sin embargo, las animaciones que he mencionado siguen entre los archivos del juego.

De todas formas, estas ideas no fueron descartadas del todo. La persecución en un escenario infinito a modo de pesadilla, fue implementada de forma más sencilla en la introducción del juego (ya hablaremos de eso más tarde). Y la batalla “shooter” en un tablero de ajedrez si fue llevado a cabo, pero con algunas modificaciones. Veamos cómo fue su desarrollo y cómo quedó.

El tablero de ajedrez se disminuyó drásticamente para que la luz del enemigo rebotase contra más elementos, y quedase mejor. La cámara cambió de tercera persona a primera, con su control correspondiente (este es el único script que yo no he realizado en todo el proyecto (ojo, el disparo sí lo he hecho yo)).



Dependiendo de la pieza de ajedrez, tiene su salud y animación (lo que hace que su “ataque” sea distinto). Por ejemplo, las torres son las más duras. Su ataque/animación es el más difícil de esquivar, y, además, requiere del mayor número de balas para derrotarlas. Los

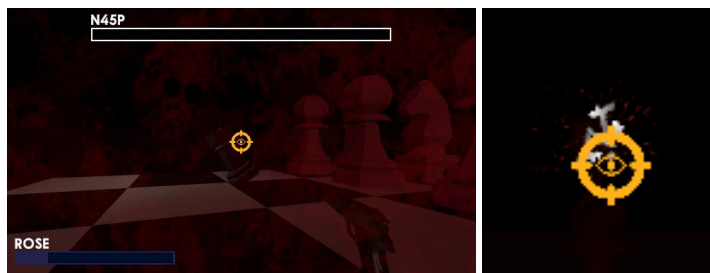
caballos van de adelante hacia atrás, los alfiles rotan, los peones son simples.

Cada pieza tiene un límite máximo de apariciones, por ejemplo, solo puede haber dos torres en el tablero, ya que son las más complicadas.

El jugador puede disparar a las piezas para provocarles daño hasta derrotarlas. Emiten un sonido y unas partículas.

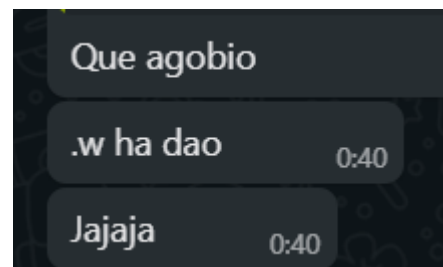


Como se aprecia, el jugador tiene su propia barra de vida y el enemigo principal otra. Si el jugador recibe daño, emite un sonido y un efecto de sangre en pantalla. Si el enemigo recibe daño, expande unas partículas “eléctricas” junto a un sonido.



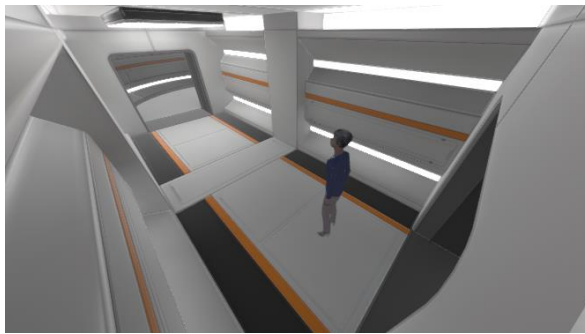
Algo en lo que me he centrado mucho, es dar un “feedback” del combate, es decir, dar la sensación de que de verdad estás disparando a fichas de ajedrez. Y la verdad que, gracias a las animaciones, sonidos y demás, lo he conseguido. Derrotar a esas malditas piezas de ajedrez me parece muy satisfactorio.

Quería que no fuera un combate de acción, sino más bien, una experiencia agobiante y claustrofóbica. Y creo que con la iluminación, escenario, enemigos, y demás, lo he conseguido. Pasé un vídeo a unos amigos de cómo estaba yendo la cosa, sin decir lo que buscaba, y uno lo dijo:





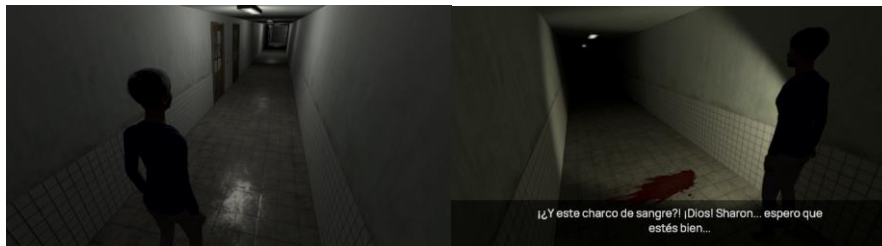
Cuando el jugador muere, salta una escena de muerte perturbadora. En cambio, si el jugador derrota al enemigo, salta otra escena, y envía al jugador a la siguiente zona: El laboratorio.



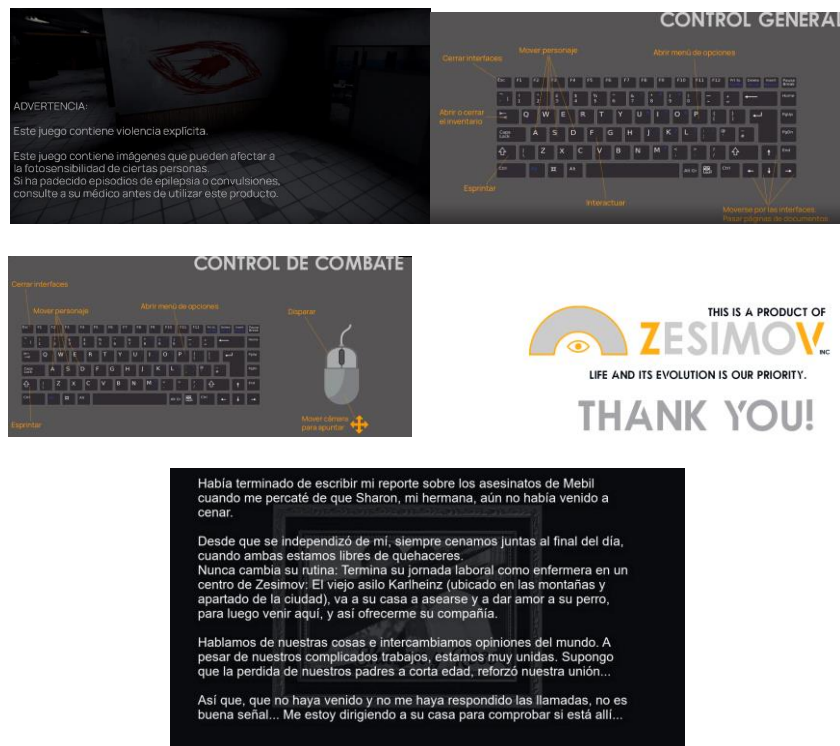
Este escenario es muy simple, y no hay nada que hacer. Solamente sirve para terminar el juego con un “continuará...”, dando a entender que la continuación se desarrollaría en este ambiente (con una jugabilidad similar al nivel del

asilo). Este escenario lo desarrollé justo antes del escenario del enemigo.

Para finalizar con la creación de los escenarios, creé una escena inicial en la que reutilicé la idea de la persecución del enemigo. Sería una escena introductoria para meter al jugador en el ambiente del juego. Primero parece un asilo normal y corriente, y cuanto más va avanzado, se irá dando cuenta de que hay algo mal en todo esto. Se encuentra sangre, las luces se apagan (teniendo que encender la linterna), y poco después, aparece el enemigo, por lo que, el jugador debe de huir en un pasillo que parece infinito, hasta que al final, es alcanzado, provocando su desmayo... Luego despierta en el asilo, y empieza el juego en sí.



Teniendo la estructura del juego hecha, creé una presentación con unas advertencias para la salud y seguridad, controles, una especie de anuncio de la empresa malvada del videojuego, y una “cinemática” inicial de dos minutos que cuenta el principio de la trama.



Había terminado de escribir mi reporte sobre los asesinatos de Mebil cuando me percaté de que Sharon, mi hermana, aún no había venido a cenar.

Desde que se independizó de mí, siempre cenamos juntas al final del día, cuando ambas estamos libres de quehaceres. Nunca cambia su rutina. Termina su jornada laboral como enfermera en un centro de Zesimov. El viejo asilo Karlheinz (ubicado en las montañas y apartado de la ciudad), va a su casa a asearse y a dar amor a su perro, para luego venir aquí, y así ofrecerme su compañía.

Hablamos de nuestras cosas e intercambiamos opiniones del mundo. A pesar de nuestros complicados trabajos, estamos muy unidas. Supongo que la pérdida de nuestros padres a corta edad, reforzó nuestra unión...

Así que, que no haya venido y no me haya respondido las llamadas, no es buena señal... Me estoy dirigiendo a su casa para comprobar si está allí...

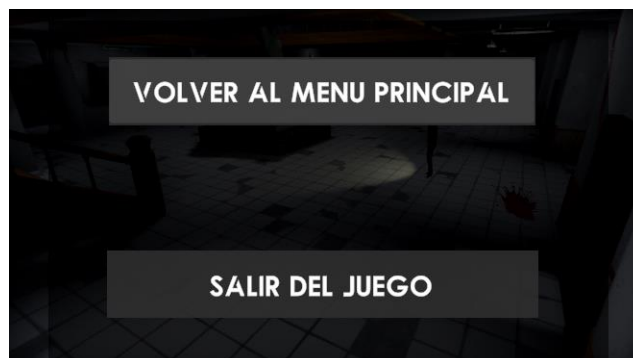
Por lo que la estructura completa del videojuego es la siguiente:

Presentación – Menú principal – Cinemática inicial – Escenario introductorio – Asilo – Batalla contra el enemigo en el tablero de ajedrez – Laboratorio.

En algún momento del desarrollo, implementé copias de seguridad en un repositorio de Github (privado). Antes de esto, realizaba copias de seguridad comprimiendo la carpeta del proyecto y guardándola en un sitio. Algo muy cutre. Pero en mis prácticas en la FCT, di un curso de Git, y entonces, me animé a implementarlo en mi juego.

0.24.5 - Se ha añadido una pantalla de muerte y una pantalla de presentación del juego.	19 may. 2021 14:39	garquesj <jokenloc	1468
0.24.4 - Se han añadido más enemigos (piezas de ajedrez) junto a sus dos colores. Se han añadido animaciones (cosa que afecta a la jugabilidad, pues por ejemplo, la torre es más difícil de esquivar). Otras mejoras. Se está trabajando en equilibrar la dificultad.	16 may. 2021 0:57	garquesj <jokenloc	2414
0.24.3 - Se ha añadido partículas específicas para cada enemigo cuando reciben daño. Se ha aumentado la altura del jugador para que no esté tan cerca del suelo.	15 may. 2021 13:05	garquesj <jokenloc	7897
0.24.2 - Se ha aumentado el tamaño del escenario (tablero), lo que conlleva muchos cambios. Se ha añadido una barra de vida al enemigo principal (la luz, 14459). Se han refactorizado ciertas cosas. Se ha mejorado otros aspectos.	14 may. 2021 20:11	garquesj <jokenloc	0640
0.24.1 - Se ha añadido la interacción y animación del daño al tocar a los enemigos (aún falta la muerte). Se ha añadido una animación de movimiento al "cuerpo" de la luz enemiga. Se ha añadido una animación de un destello (a modo de ataque de la luz).	13 may. 2021 19:00	garquesj <jokenloc	4751
0.23.6 - Se ha perfeccionado y completado el sistema de disparo, añadido animaciones, enemigos, sonidos, y demás. Falta adaptar el escenario y enemigos para una mejor jugabilidad. También se quiere añadir más enemigos y animaciones. Ahora el cañón puede disparar.	13 may. 2021 16:34	garquesj <jokenloc	4817
0.23.5 - Se está adaptando todo para realizar un combate shooter. Se está realizando el tema del disparo.	12 may. 2021 20:51	garquesj <jokenloc	5197
0.23.4 - Se ha completado lo anterior. Se han quedado varias cosas a medias porque se va a pensar una forma alternativa de realizar todo esto.	4 may. 2021 20:47	garquesj <jokenloc	4768
v0.23.3 - Se han añadido nuevas animaciones y assets. Se ha modificado el tamaño del tablero y sus "accesorios", ya que se ha cambiado el planteamiento del enfrentamiento con el enemigo. Ya no será una persecución, sino una batalla en un tablero.	3 may. 2021 21:24	garquesj <jokenloc	ba7a
v0.23.2 (Hotfix) - Se ha realizado un reset (hard) al commit 0.22.3 para pegar todos los cambios que se habían realizado (commit 23.1 y 23.2) y entonces hacer este commit. Esto se debe, a que se había introducido un asset que superaba el tamaño máximo de Github.	2 may. 2021 21:54	garquesj <jokenloc	a463
0.22.8 - Se ha terminado la parte del laboratorio. Se ha empezado a diseñar sangre por el escenario.	20 abr. 2021 21:01	garquesj <jokenloc	8807
v0.22.7 - Se ha creado la colocación de los grifos en el estatus. La estatua se mueve y deja un hueco con escaleras que llevan a una nueva escena de un laboratorio donde al caminar un poco por el mismo, finalizará el juego (esto aún está por hacer) para volver al menú principal.	20 abr. 2021 13:08	garquesj <jokenloc	d032
v0.22.6 - Se ha corregido un "bug" de lo anterior. Se han añadido unos documentos de historia y para averiguar el código de la caja de seguridad de la sala de máquinas.	19 abr. 2021 14:40	garquesj <jokenloc	b664
v0.22.5 - Se ha creado un menú de opciones dentro del juego para volver al menú principal o salir del juego.	19 abr. 2021 11:55	garquesj <jokenloc	6648
v0.22.4 - Se ha mejorado el puzle de la baldosa secreta en los dormitorios para que sea menos trivial y más inmersivo en la historia.	16 abr. 2021 12:24	garquesj <jokenloc	e9ba
v0.22.3 - Se ha introducido un susto al intentar recoger la tarjeta correcta. Se han realizado mejoras. Hay pendiente de corregir un bug en los mensajes.	16 abr. 2021 0:02	garquesj <jokenloc	b3ba
v0.22.2 - Se ha completado el sistema de las tarjetas, escape de gas y caja de seguridad (falta añadir documentos que indiquen la contraseña). Se está trabajando en un "jumpscare" para cuando se vaya a coger la tarjeta correcta.	14 abr. 2021 11:10	garquesj <jokenloc	a068
v0.22.1 - Se ha realizado una parte del sistema de tarjetas de identificación para la caja de seguridad.	14 abr. 2021 13:09	garquesj <jokenloc	b499
v0.21.2 - Se ha completado lo anterior hasta tener la obtención del segundo grifo. Se ha empezado a hacer el puzle de las tarjetas de identificación. Se ha añadido la caja de seguridad que contiene el tercer grifo en la sala de máquinas.	13 abr. 2021 20:16	garquesj <jokenloc	6a60
v0.21.1 - Se ha realizado la interacción del código de la pizana. Falta añadir la caja de seguridad (y lo demás) correspondiente a dicho código. Por las cámaras fijas, se ha eliminado los audio listener de las cámaras, y se ha añadido uno al jugador. Se han añadido más animaciones.	12 abr. 2021 22:23	garquesj <jokenloc	49c8
v0.20 - Se ha trabajado en el puzle de la mancha negra. Se ha refactorizado el código de la caja de seguridad. Se han realizado otras mejoras. Se ha empezado a realizar el puzle de la pizana.	10 abr. 2021 20:57	garquesj <jokenloc	244a
v0.19 - Se ha realizado la animación de la mancha negra y la animación de la mancha roja. Se ha añadido la animación de la mancha azul. Se ha añadido la animación de la mancha amarilla. Se ha añadido la animación de la mancha verde. Se ha añadido la animación de la mancha morada. Se ha añadido la animación de la mancha naranja. Se ha añadido la animación de la mancha púrpura. Se ha añadido la animación de la mancha rosa. Se ha añadido la animación de la mancha gris. Se ha añadido la animación de la mancha blanco. Se ha añadido la animación de la mancha negro. Se ha añadido la animación de la mancha rojo. Se ha añadido la animación de la mancha azul. Se ha añadido la animación de la mancha verde. Se ha añadido la animación de la mancha morada. Se ha añadido la animación de la mancha naranja. Se ha añadido la animación de la mancha púrpura. Se ha añadido la animación de la mancha rosa. Se ha añadido la animación de la mancha gris. Se ha añadido la animación de la mancha blanco. Se ha añadido la animación de la mancha negro.	9 abr. 2021 20:16	garquesj <jokenloc	3a4a

Otra cosa que no he comentado es que, incluí un menú de opciones en todos los escenarios del juego, para que el jugador pueda salir del juego o volver al menú principal.



## Conclusión

Sinceramente estoy muy contento con el resultado. La jugabilidad e historia me han encantado. He cumplido casi todos los objetivos impuestos, excepto el tema del enemigo (que ha cambiado) y el inventario (aunque realmente no ha sido necesario hacerlo complejo, ni darle un límite, ni nada de lo que planteé en un inicio). Todo lo he hecho yo, a excepción de los la mayoría de los assets, algunas animaciones, y el script del control en primera persona que se utiliza en el escenario del combate. Además, considero que el juego está pulido, creo que será difícil encontrar bugs. Eso sí, he notado que, en equipos de pocos recursos, a veces, hay un bug de sombras (cosa que a veces me ocurría al reproducirlo en el editor de Unity). No sé si funcionará del todo bien en este tipo de equipos...

Me hubiera gustado implementar más ideas que se me ocurrieron, aunque yo sabía que no me iban a dar tiempo:

- Más documentos de historia (algunos en cajones ocultos).
- Un puzle de forzar una cerradura cada vez que se requiere la ganza.
- Más enemigos (los zombis robóticos que mencioné)
- Un minijuego en las puertas con cerradura en pieza de ajedrez. En vez de que simplemente salga un texto indicando de que tipo de puerta es, habría un marco al lado de cada puerta en la que, contendría una partida de ajedrez en curso, junto a un posible jaque mate en un solo movimiento. Si el jugador descubría que pieza haría el jaque mate, perfecto, si no, se activaría una alarma, y aparecerían enemigos.
- Más sustos.
- Un “puzle” para extraer una llave de un trozo de carne muy asqueroso.
- El permitir guardar y cargar la partida. Para mí es la mayor carencia del juego, pero en el caso de este juego, requiere mucho trabajo (cosas como “guardar” que una puerta esté abierta o cerrada, por ejemplo) y no me dio tiempo.
- Etc.

No descarto seguir trabajando en él, y quien sabe, quizás (cambiando músicas y cosas que tienen copyright), publicarlo de forma gratuita en foros o similares.

Pero ya digo, estoy muy contento con el resultado.



## Créditos

Videojuego realizado con **Unity 3D** creado por Unity Technologies.

### **Creador (Dirección, programación, guion, etc.)**

Gonzalo Arques

### **Diseño del nivel del asilo**

Nacho.M  
Gonzalo Arques

### **Ayuda a la toma de decisiones**

Rafa.M  
Nacho.M

### **Beta testers**

Nacho.M  
Rafa.M  
Aaron. M  
Andrea. C  
María. C

### **Música**

Resident Evil Village, Silent Hill 1, Outlast, Resident Evil 2 (2019), Oxygen

### **Sonidos**

La mayoría de los sonidos pertenecen a Resident Evil 3 (1999), extraídos por “Roger (PP)”, y el resto son sacados de distintos sitios de Youtube.

### **Programas utilizados**

Unity 3D  
Adobe Photoshop  
Camtasia Studio 9  
Audacity

### **Webs utilizadas**

Youtube

Mixamo

audio.online-convert.com

Foros de ayuda de Unity, Stackoverflow...

### **Assets de la Unity Asset Store**

**PBR Chess Key / Fuse Model Pack**

**Abandoned Asylum**

**Free Low Poly Chess Set**

**3D Scifi Kit Starter Kit**

**Double Open End Wrench**

**Pack of five optimized PBR low-poly models of a griffins**

**Fire Extinguisher**

**3D Scifi Kit Starter Kit**

**Morgue Room PBR**

**LowPoly Sci-Fi Crates Free**

**JukeBox Music Player**

**Tube constructor kit**

**Tool Icons and Blood Sprites**

**Low Polygon Potions Pack**

**PBR - Hospital Horror Pack. Free**

**Allen Key Set - PBR**

**Modern Guns: Handgun**

**FREE Floral Gold Jar**

**Standard Assets (for Unity 2018.4)**