



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN DESARROLLO DE
VIDEOJUEGOS Y REALIDAD VIRTUAL

**Comparación entre mapas Generados
Proceduralmente y mapas con Diseño Predefinido
para juegos de terror**

DIEGO VALDERRAMA

Profesor Guía: LIZA JEGÓ

Memoria para optar al título de
Ingeniero en Desarrollo de Videojuegos y Realidad Virtual

Talca – Chile
Junio, 2025



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN DESARROLLO DE
VIDEOJUEGOS Y REALIDAD VIRTUAL

**Comparación entre mapas Generados
Proceduralmente y mapas con Diseño Predefinido
para juegos de terror**

DIEGO VALDERRAMA

Profesor Guía: LIZA JEGÓ

Memoria para optar al título de
Ingeniero en Desarrollo de Videojuegos y Realidad Virtual

El presente documento fue calificado con nota: _____

Talca – Chile

Junio, 2025

*Dedicado a Lisbeth González, Paula Pérez, Alberto Valderrama y mi familia y
amigos por apoyarme todos estos años.*

AGRADECIMIENTOS

Agradecimientos a Lisbeth González, mi pareja, por estar siempre conmigo y apoyándome, pasando tardes juntos trabajando y jugando. También a mis compañeros de carrera, los cuales me han ayudado bastante estos últimos años, con los cuales he compartido tanto de manera presencial como de manera online.

Igualmente, quisiera agradecer a Paula Pérez y Alberto Valderrama, mis padres, por estar ahí para mí cuando los necesitaba y también a mis familiares, quienes confían en mí, incluyendo a mi perro Zeke, que a pesar de que no lo cuido tan bien, me quiere mucho.

Finalmente, quisiera agradecer a todos mis otros amigos, la mayoría de la escuela, con los cuales he compartido mucho y, a pesar de quizá no hablar mucho con ellos últimamente, sé que me han apoyado y estoy muy feliz por ello.

TABLA DE CONTENIDOS

	página
Dedicatoria	I
Agradecimientos	II
Tabla de Contenidos	III
Índice de Figuras	V
Índice de Tablas	VI
Resumen	VII
 1. Introducción	 1
1.1. Problema	2
1.2. Objetivos	3
1.2.1. Objetivo General	3
1.2.2. Objetivos Específicos	3
 2. Estado del Arte	 4
2.1. Lethal Company	4
2.2. Content Warning	6
2.3. Phasmophobia	7
 3. Marco Teórico	 8
3.1. Juegos de terror	8
3.1.1. Mapas prediseñados	9
3.1.2. Mapas generados proceduralmente	10
 4. Metodología	 12
4.1. Metodologías de desarrollo de software	12
4.2. Metodologías de obtención de datos	13
4.3. Carta Gantt	14

5. Implementación	15
5.1. Desarrollo de salas	15
5.2. Wave Function Collapse	17
6. Resultados	23
6.1. Aplicación del experimento	23
6.2. Generales	25
6.3. Mapas prediseñados	26
6.4. Mapas generados proceduralmente	29
6.5. Apartado Final	32
7. Conclusiones	33
7.1. Trabajo futuro	33
Bibliografía	35
A: Anexos	39
A.1. Carta Gantt	39
A.2. GDD	39
A.3. Video Gameplay	39
A.4. Cuestionarios	39
A.5. Consentimiento Informado	49

ÍNDICE DE FIGURAS

	página
2.1. Interior de la nave	5
2.2. Mapa de la Luna Experimentation	5
2.3. Mapas del juego Content Warning	6
2.4. Mapa 10 Ridgeview Court	7
5.1. Salas en forma de L	16
5.2. Las otras Salas	17
5.3. Mapas generados proceduralmente por el algoritmo	22
6.1. Pruebas con usuarios	24
6.2. Cuan frecuente juegan juegos de terror	25
6.3. Nivel de miedo en mapas prediseñados	26
6.4. Qué tan intuitivos fueron los mapas prediseñados	27
6.5. Qué tan equilibrados fueron los mapas prediseñados	28
6.6. Nivel de miedo en mapas generados proceduralmente	29
6.7. Qué tan intuitivos fueron los mapas generados proceduralmente . . .	30
6.8. Qué tan equilibrados fueron los mapas generados proceduralmente . .	31
6.9. Qué tan rejugables hicieron los mapas generados proceduralmente al juego.	31

ÍNDICE DE TABLAS

página

RESUMEN

El género de terror en los videojuegos usa varias formas de dar miedo al jugador, tales como gráficos que generen ambientes o criaturas espeluznantes o audios y sonidos que aumenten el miedo en el jugador mediante sonidos de diferentes calidades. Una de las formas que más se usa es en cómo es diseñado y creado el mapa del juego, permitiendo al jugador experimentar diversos niveles de tensión o temor dependiendo de en qué se enfoque el desarrollador a la hora de crear un nivel. Últimamente, se han popularizado bastante los juegos de terror, en donde el jugador es mandado a una locación donde tiene que conseguir ciertas cosas u objetos antes de poder irse del lugar. En estos juegos se tiende a utilizar 2 formas de creación de niveles que se destacan por sobre el resto: la generación procedural y el diseño a mano o mapas con diseño predefinido, que es como se les referirá en esta memoria; ambas formas tienen sus ventajas y desventajas a la hora de implementarlas para generar miedo en un jugador.

La generación procedural, por ejemplo, es un método de creación de mapas que utiliza un algoritmo para crear niveles o mapas de manera automática y con un potencial de creación infinito. Este método se utiliza para varios géneros de videojuegos, pero en videojuegos de terror se utiliza bastante para generar incertidumbre en lo que parece un espacio infinito, creando salas parecidas unas entre otras, dando un sentimiento de déjà vu o de salas que no parecen tener propósito. Por otro lado, están los mapas con diseño predefinido; estos son mapas que se crean a mano, permitiendo tener un mayor control al desarrollador sobre la narrativa, estética y jugabilidad, dándoles la oportunidad de generar momentos específicos de mayor temor o tensión, al igual que poder contarle al jugador la historia de su mundo a través del apartado visual del nivel o de hacer la dificultad progresiva, permitiendo al jugador aprender y evolucionar con la progresión del nivel.

Esta memoria tiene como objetivo el desarrollo de un juego del género de terror y supervivencia en el que el jugador va a tener el objetivo de buscar una cantidad específica de objetos dentro de un mapa generado proceduralmente o uno con diseño predefinido, en donde se podrán comparar los niveles de miedo que pueden generar estos 2 mapas a través de 2 niveles y poder determinar cuál es más eficaz a la hora de dar miedo a los jugadores. Lo que se busca conseguir principalmente es determi-

nar que los mapas generados de manera procedural son más efectivos a la hora de generar miedo que los mapas con diseño predefinido; esto se conseguirá a través de un cuestionario que se le hará al jugador después de superar por primera vez el nivel y posteriormente cuando supere por segunda vez el nivel, ambos cuestionarios serán iguales en tipos de pregunta, pero la segunda poseerá más preguntas que la primera, se les preguntará por cosas como su experiencia con el nivel, como se sintió al jugarlo de nuevo, si es que se le hizo más fácil la segunda vez y se usará la herramienta *Subjective Units of Distress Scale* para obtener sus niveles de angustia y ansiedad. Se espera también que de esta forma se le pueda dar información al lector sobre cómo funcionan estos 2 tipos de mapa, tanto sus diferencias como similitudes, a la hora de crear un juego del género de terror, para poder determinar correctamente qué tipo de mapa usar dependiendo del enfoque de su juego.

1. Introducción

Los juegos del género de terror remontan sus orígenes a finales de 1970 e inicios de 1980, con juegos como “Uchu Yusosen Nostromo” [1], considerado como el primer juego de terror y supervivencia lanzado para la serie AX de ASCII, el cual tiene como inspiración la película de Alien de 1979. Este género seguiría evolucionando con clásicos como “Alone in the Dark”(1992), “Clock Tower”(1995) o “Resident Evil”(1996), llegando a juegos más modernos como “Until Dawn”(2015), “Tormented Souls”(2021) o “Alan Wake 2”(2023). Este tipo de juegos tiene muchos elementos que los hacen reconocibles; entre ellos está el uso de una atmósfera con iluminación tenue y entornos muy oscuros, acompañados por una música tensa que se intensifica cuando ocurren momentos peligrosos para el jugador. También se usa bastante el que el jugador posea recursos limitados como de salud o una linterna a la que se le tenga que ir buscando baterías, lo que llevaría a la exploración, hacer que el jugador se mueva por un mapa con enemigos acechándolo, esperando el momento perfecto para asustarlo [2]. El mapa de juego es uno de los aspectos más importantes dentro de todo tipo de juego y comúnmente en juegos de terror se tiende a usar mucho los mapas prediseñados, mapas hechos completamente a mano por un diseñador, donde ellos controlan los diversos aspectos del juego, como la atmósfera, los recursos, enemigos o mecánicas que se van a usar en ese momento, pudiendo incluso guiar al jugador hacia donde ir y qué hacer resaltando caminos u objetos o incluso bloqueando posibles caminos [3]. Con el avance del tiempo, se ha empezado a usar más y más la generación procedural para la creación de estos mapas. Esto se hace a través de un algoritmo de generación automática de contenido al cual se le otorgan parámetros para que genere los que uno desea, reduciendo costos de producción y otorgando control paramétrico sobre los resultados [4].

En el capítulo 2 se habla sobre el estado del arte de los videojuegos de terror y supervivencia, mostrando ejemplos de videojuegos recientes que utilizan ambos métodos de creación de niveles; en el capítulo 3 se hablará sobre algoritmos que se usarán en el proyecto, al igual que especificaciones sobre las herramientas que se usarán para desarrollar el videojuego, en el capítulo 4 se explicaran las metodologías que se aplicaran durante el desarrollo del software al igual que para la obtención de datos.

1.1. Problema

El miedo es un factor importante para los juegos de terror, y en los últimos tiempos se han popularizado los juegos de terror y supervivencia cooperativos, donde el jugador junto a un grupo de amigos juega varias veces un mismo nivel, haciendo que sea fundamental saber balancear los aspectos del juego para permitir una buena rejugabilidad [5]. Uno de los aspectos más relevantes para la rejugabilidad es el mapa o nivel de juego, debido a que ahí es donde el jugador va a pasar la mayor cantidad de tiempo. Comúnmente, dentro del género de terror se tiende a utilizar más los mapas prediseñados. Este tipo de mapas, aunque bueno para guiar al jugador, tiende a afectar negativamente la rejugabilidad debido a que no presenta nada nuevo para el jugador. Tras un tiempo de ir jugando, el usuario se empezará a acostumbrar a la disposición del nivel y a cuando ocurren los momentos de miedo, bajando sus niveles de miedo y haciendo que todo el proceso sea un momento más tedioso que de miedo. En contraparte están los mapas generados proceduralmente. Este tipo de mapas son generados a partir de un algoritmo y, aunque el diseñador tiene un menor control sobre el resultado, este algoritmo puede ser ajustado para conseguir un resultado de mapa óptimo. Este tipo de mapas también tiene la ventaja de no ser el mismo para todos los jugadores, ya que se genera uno nuevo cada vez que se entra al mapa, aumentando grandemente la rejugabilidad, haciendo que el jugador no se pueda acostumbrar a un mismo mapa y tenga que estar más atento de sus alrededores, haciéndolo más susceptible a momentos de miedo.

Para solucionar el problema de los mapas prediseñados, se va a desarrollar un videojuego del género de terror y supervivencia donde se apliquen ambos métodos de creación de niveles, tanto el prediseñado como el de generación procedural, para medir los niveles de miedo de los jugadores. Esto se refiere a las emociones que sentirá

el jugador, medir qué tan ansioso, nervioso o angustiado se siente el jugador después de la experiencia de juego. Esto se va a medir mediante un estudio principalmente cualitativo en forma de cuestionario que se le hará a los jugadores tras jugar el juego [6]. En este cuestionario, se les preguntará sobre los niveles de miedo que sintieron al jugar el juego, el sentimiento de desorientación que experimentaron, en qué afectó el diseño del nivel con respecto a su experiencia de juego y su opinión sobre rejugabilidad. Estos datos servirán para verificar la hipótesis: Un mapa generado proceduralmente provoca más niveles de miedo al jugador que un mapa con diseño predefinido.

1.2. Objetivos

En esta sección se presentará el objetivo general, al igual que los objetivos específicos que se deben completar.

1.2.1. Objetivo General

- Evaluar si los mapas generados de manera procedural son más efectivos para generar miedo en comparación con los mapas de diseño predefinido en un juego de terror.

1.2.2. Objetivos Específicos

1. Desarrollar un videojuego del género de terror, utilizando como lenguaje de programación C# y Unity como plataforma de desarrollo.
2. Desarrollar un algoritmo procedural para la creación de un nivel.
3. Desarrollar un nivel que no posea el algoritmo procedural para hacer comparaciones.
4. Crear un cuestionario para obtener la opinión de los usuarios con respecto al miedo que sintieron al jugar dos niveles.
5. Analizar los datos de los cuestionarios para verificar si se cumple o no la hipótesis de esta investigación.

2. Estado del Arte

En este segundo capítulo se hablará de juegos del género de terror que utilizan alguno de los métodos de creación de niveles y cómo los implementan para la atmósfera del juego.

2.1. Lethal Company

Lethal Company [7] es un juego de terror y supervivencia, desarrollado y publicado por Zeekerss en la plataforma de Steam como un juego de acceso anticipado el día 23 de octubre de 2023. Lethal Company es un juego cooperativo de hasta 4 jugadores, donde juegas como un trabajador de “La Compañía” y tu trabajo es el de recolectar chatarra de diferentes lunas que fueron abandonadas, para luego venderlas y cumplir con una cuota que va aumentando a medida que avances en el juego. Dentro del juego, el jugador cuenta con un tiempo límite, donde tendrá que investigar las locaciones en las lunas en búsqueda de chatarra y sobrevivir a los enemigos que merodean tanto por fuera como por dentro de las instalaciones. El gameplay del juego se desarrolla en 3 locaciones, la nave, la luna y la instalación. Estas 2 primeras, como se pueden ver en la figura 2.1 y la figura 2.2, están hechas con un diseño predefinido, mientras que la última, y la que más se va a utilizar como inspiración, está hecha de manera procedural utilizando diferentes piezas para crear el mapa, haciendo de que, a pesar de visitar una misma luna repetidas veces, esta no sea la misma en su interior.



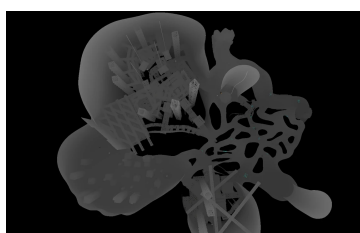
Figura 2.1: Interior de la nave



Figura 2.2: Mapa de la Luna Experimentation

2.2. Content Warning

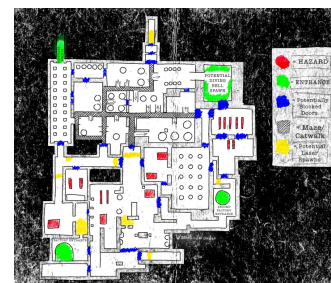
Content Warning [8] es un juego cooperativo de terror y supervivencia desarrollado por thePetHen, Skog, Zorro, Wilnyl, Philip y publicado por Landfall Publishing en la plataforma de Steam el día 1 de abril de 2024. En Content Warning, el jugador forma parte de un equipo de grabación donde él y hasta 3 compañeros más deben aventurarse a “El Viejo Mundo” para grabar la mayor cantidad de monstruos y contenido terrorífico que puedan, para posteriormente subirlo a una plataforma llamada “SpöökTube” y de esta forma conseguir la meta de visitas y ganar dinero que se puede gastar en mejoras de equipo y otras cosas que permitan generar mayor cantidad de visitas. El gameplay del juego se ubica en 2 locaciones, la isla principal, siendo este el lobby del juego, donde los jugadores descansan, pueden conseguir nuevo equipo para los videos y subir los videos grabados, y “El Viejo Mundo”, que es donde los jugadores tienen que grabar el contenido. Como se puede ver en la figura 2.3, este lugar se divide en 3 mapas con un diseño predefinido, aunque pueden tener ciertas modificaciones. Estos mapas son elegidos de manera aleatoria, pero se mantiene la elección hasta que se cumpla o falle la meta de visitas. En el juego, el jugador cuenta con 3 días para cumplir la meta de visitas y cuando se cumple se le asigna una nueva y más alta meta. Igualmente, dentro del viejo mundo, al jugador se le limita la visita de 2 maneras diferentes, siendo una el oxígeno de cada jugador y la capacidad de grabación de la cámara. Ninguno de estos se puede recuperar o extender, haciendo que los jugadores tengan que aprender a utilizar de buena manera estos recursos para lograr volver a la superficie y publicar el video del día.



(a) Minas



(b) Barco



(c) Fábrica

Figura 2.3: Mapas del juego Content Warning

2.3. Phasmophobia

Phasmophobia [9] es un juego de terror y supervivencia cooperativo, el cual fue desarrollado y publicado por Kinetic Games como un juego con acceso anticipado el 18 de septiembre de 2020 en la plataforma de Steam. En este juego, el jugador toma el papel de un investigador paranormal donde él y hasta otros 3 compañeros deben buscar evidencia de actividad paranormal en locaciones embrujadas y de esta forma identificar correctamente el tipo de fantasma según la evidencia que se recolecte; para esto los jugadores poseen diversas herramientas. Dentro del juego, el jugador se puede mover por 2 ubicaciones: el lobby donde los jugadores pueden escoger qué misión hacer y donde puedes comprar las diferentes herramientas y mejoras de estas para utilizarlas en las misiones; también están las locaciones, las cuales son 10 mapas diferentes, los cuales todos poseen un diseño predefinido. Como ejemplo, la figura 2.4 muestra el mapa 10 Ridgeview Court. El hecho de que el mapa sea algo estándar permite al jugador familiarizarse con las locaciones y así poder hacer estrategias según el lugar donde vayan.

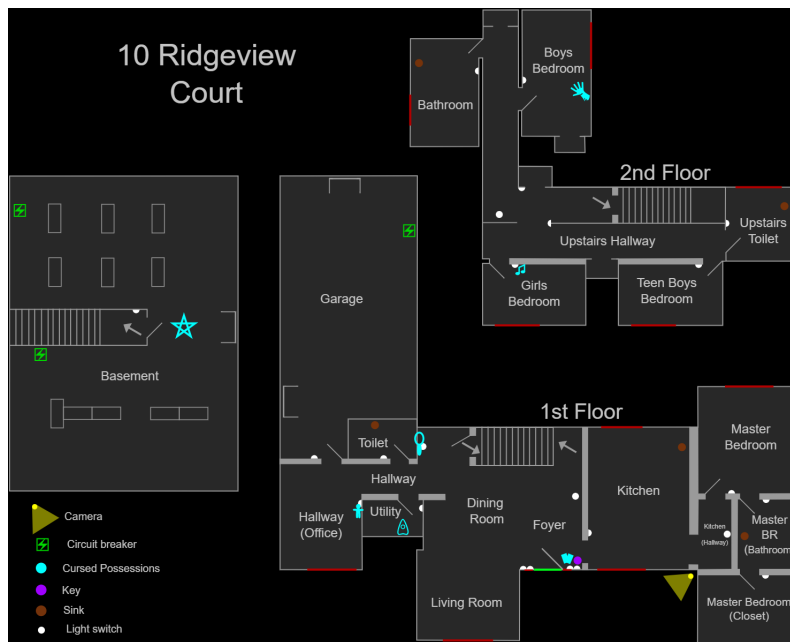


Figura 2.4: Mapa 10 Ridgeview Court

3. Marco Teórico

3.1. Juegos de terror

Los juegos de terror utilizan diversos medios y métodos para provocar miedo en el jugador, utilizando la atmósfera del juego, su apartado visual e incluso el apartado sonoro en conjunto con el gameplay para crear una experiencia constante de terror, utilizando estos elementos lúdicos [10] para hacer más partícipe al jugador de la experiencia; estos elementos son parte del diseño del videojuego, cosas como la apariencia de los enemigos, cómo se comportan estos mismos y uno de los apartados más importantes, el mapa de juego. El mapa del juego [11] es bastante importante a la hora de hablar de cómo un juego puede guiar a sus jugadores; esto se debe a que, a través de los elementos que posee, obliga al jugador a adaptarse, a posicionarse de diferentes formas que quizá no sean convencionales, a moverse por caminos estrechos o preocuparse debido a lo amplio y vacío del terreno; incluso puede resaltar mecánicas nuevas o familiares para el jugador, todo esto gracias al diseño del mapa de juego. Comúnmente los mapas se crean a mano; se dedica parte del desarrollo a crear los mapas, a optimizarlos y a posicionar los elementos correctos para que el jugador experimente ciertas emociones en el momento exacto, haciendo que los diseñadores posean control casi total sobre las emociones del jugador, aunque esto viene con la carga de tener que dedicarles el tiempo necesario, provocando posibles retrasos a la hora de terminar de crear el videojuego. En la actualidad, más y más se han empezado a utilizar algoritmos para facilitar y agilizar el diseño de niveles [12], haciendo el trabajo más fácil, pero aun dando cierto nivel de control al diseñador para poder corregir y modificar los diseños que otorgan los algoritmos. Igualmente, existen desarrolladores que adoptan completamente el uso de algoritmos y su optimización para lograr crear mapas sin la necesidad de modificarlos. El uso de la

generación procedural para lograr mapas casi infinitos sin tanto esfuerzo proporciona a los desarrolladores más tiempo que se puede enfocar en otras cosas, algoritmos que se basan en lo inesperado para otorgarle a los jugadores experiencias únicas cada vez que juegan, perdiendo el enfoque narrativo que otorga el control total del diseño de nivel en pos de generar mayor incertidumbre en el jugador.

3.1.1. Mapas prediseñados

El diseño de mapas es algo principal en el desarrollo de un videojuego, y es muy común el hacerlos de forma manual, casi siempre iniciando con un boceto para después crear los elementos del mapa y ordenarlos de la mejor forma posible para poder presentarle al jugador un mundo. Este método otorga a los diseñadores un control absoluto de la atmósfera del juego e incluso ayuda a poder contar una historia únicamente con aspectos visuales, guiando al jugador y mostrándoles parte de la historia del juego [13]; es muy bueno igualmente para poder controlar ciertos eventos del juego, pudiendo colocarlos donde más se desea para maximizar el impacto emocional que estos puedan provocar.

Al momento de generar miedo, un mapa prediseñado ayuda mucho a crear ciertos momentos en los que se quiere asustar al jugador, diseñar secuencias claustrofóbicas o hacer sentir diminuto al jugador en comparación a sus enemigos. Aunque esto no viene con sus consecuencias, este estilo de creación de mapas limita mucho la rejugabilidad de un juego, provocando que un jugador, con el tiempo, se aprenda dónde tiene que ir o qué es lo que va a ocurrir después, quitando esa sensación de lo inesperado que viene con la primera impresión.

A diferencia de los mapas generados proceduralmente, estos mapas son resultado de un proceso creativo humano, con el enfoque de crear desafíos y provocar miedo en el jugador, al igual que poder contar una historia, por lo cual no se usan algoritmos, sino métodos para lograr ciertos objetivos, por ejemplo, la segmentación, para poder dividir el nivel en módulos o habitaciones interconectadas, el uso de bocetos en papel para poder hacer varios diseños rápidos para después tomar el mejor o buscar un enfoque narrativo usando un diseño intencional o un enfoque psicológico, usando principios de psicología para diseñar mapas desafiantes para el jugador.

3.1.2. Mapas generados proceduralmente

La generación procedural es un método de creación que se rige por el uso de algoritmos que le den órdenes mediante parámetros ingresados por los programadores para generar el contenido, reemplazando el trabajo manual de un diseñador. Esto trae sus ventajas y desventajas, por ejemplo, la reducción de costos de desarrollo y la mayor rejugabilidad que le puede otorgar a un videojuego, pero, en cambio, se pierde la inmersión narrativa que puede crear un mapa prediseñado y limita el control directo del producto final.

A la hora de generar miedo, el uso de algoritmos de generación procedural otorgan amplios beneficios que aunque se puedan lograr con un mapa prediseñado, se va a necesitar emplear bastante tiempo y costes de desarrollo para lograr algo parecido; el no saber que esperar, la incertidumbre que provoca el no saber que viene potencia la atmosfera de miedo que transmite el juego [14], no saber como escapar del enemigo o que al volver a jugar el camino no sea el mismo provoca que el jugador tenga que estar más pendiente, más alerta y en consecuencia más asustado o más propenso al susto, lamentablemente como se menciona con anterioridad esto afecta a la narrativa del juego, se puede ver afectado bastante el contar una historia, al igual que encontrar un balance entre dificultad y jugabilidad, pudiendo hacer al juego algo frustrante, pero que aun así no sea algo repetitivo para el jugador.

En la actualidad existen varios algoritmos, herramientas [15] o sistemas [16] para la generación procedural, facilitando un poco el trabajo del programador de necesitar el mismo algoritmo, por ejemplo:

Algoritmos evolutivos

Los algoritmos evolutivos son un tipo de algoritmos de búsqueda y optimización que utilizan conceptos de evolución natural, utilizando mecanismos como la reproducción, mutación y recombinación [17]. Este tipo de algoritmos es bastante útil cuando se quiere resolver un problema complejo y es usado en varias áreas, no solamente videojuegos o inteligencia artificial.

Los algoritmos evolutivos, en el caso de usarlos para generar mapas para juegos, son bastante útiles debido a que pueden generar múltiples variaciones, de las cuales se pueden escoger los mejores de acuerdo a ciertos criterios específicos implementados en la “fitness function” [18, 19]. Esta es una función de evaluación que sirve para medir

y controlar los operadores genéticos. Esta función también sirve para determinar los candidatos más aptos, como por ejemplo un mapa de una dungeon que contenga puertas cerradas que necesiten de misiones para lograr abrirlas [20], pudiendo usar los algoritmos evolutivos para generar amplias posibilidades, incluso en el caso de esta memoria, se puede configurar para lograr crear mapas que provoquen una mayor cantidad de miedo, usando métricas que busquen mayor ansiedad o inquietud a través de la apariencia del entorno.

Wave Function Collapse

Wave Function Collapse es un algoritmo de generación procedural, el cual genera contenido a partir de la imitación de patrones existentes. Esto lo hace siguiendo un conjunto de reglas que se le imponen para que pueda hacer conexiones coherentes [21]. El WFC en programación es una imitación al colapso de la función de onda de la mecánica cuántica, el cual habla de cómo una función de onda pasa de una superposición cuántica a estar en un simple estado debido a la interacción con un observador.

Maxim Gumin desarrolló un asset utilizando este algoritmo para la generación de bitmaps y tilemaps a partir de un simple ejemplo que se le otorgue; este asset posteriormente ha sido portado para ser usado en diferentes lenguajes de programación e incluso diferentes motores de desarrollo. Joseph Parker y BorisTheBrave desarrollaron individualmente diferentes ports para el uso en Unity.

Este algoritmo sirve bastante para la generación de mapas debido a que se le puede dar un pequeño ejemplo de cómo se espera que se vean las salas, y este posteriormente usará ese ejemplo para generar diferentes tipos de mapas, que pueden ser optimizados para generar un sentimiento de miedo gracias a que generó habitaciones muy amplias y oscuras o a que se fue por el camino de generar varios pasillos largos y estrechos, así logrando un diseño más deseado y pudiendo seguir modificando los parámetros para lograr diferentes resultados. [22].

4. Metodología

4.1. Metodologías de desarrollo de software

Dentro de los tipos de metodologías de desarrollo, existen las metodologías ágiles [23], las cuales se enfocan en poder adaptarse rápidamente a medida que van surgiendo nuevas necesidades y cambios dentro del proyecto; dentro de este tipo de metodologías existen Kanban y Scrum [24, 25]. Kanban fue diseñado por la compañía de autos Toyota para la gestión transparente del flujo de trabajo a través de la visualización del estado actual de cada tarea, dividiéndolas en “Por hacer”, “En progreso” y “Finalizado”. Por otro lado, Scrum es un enfoque ágil, iterativo e incremental, el cual utiliza bloques de tiempo de entre una semana y un mes para lograr objetivos del proyecto a través de Sprint. En este proyecto se usará una combinación de los elementos de Scrum y Kanban para lograr un trabajo fluido:

- Se desarrollará un *Product Backlog*, el cual es una lista de funciones y requisitos del producto final.
- Se van a planificar sprints, cada cierto tiempo con duraciones similares, donde se establecerá un objetivo del *Product Backlog* a alcanzar.
- Se dividirá el objetivo en tareas claras y, según prioridad, se irán moviendo en una tabla para saber su estado actual, para agilizar el trabajo.
- Al finalizar el tiempo de un sprint se realizará una revisión, donde se recibirá retroalimentación del producto actual para futuras mejoras.
- Este proyecto se realizara en conjunto con Lisbeth González, donde se trabajarán tareas comunes para el desarrollo del proyecto en conjunto, pero las tareas

correspondientes a la investigación de cada uno se desarrollaran de manera individual.

El motor de desarrollo seleccionado para realizar este proyecto es Unity, más específicamente la versión de Unity 6000.0.33f1; esta elección es principalmente por el hecho de ser una de las últimas versiones sacadas hasta el momento, al igual que posee LTS o soporte a largo plazo, refiriéndose a que es una versión estable de Unity la cual ofrece estabilidad y soporte para diferentes flujos de trabajo. Para el control de versiones se usará la herramienta de Unity Version Control, la cual está implementada directamente en Unity como parte de Unity DevOps. Esta herramienta, además de otorgar la posibilidad de guardar versiones del proyecto durante su etapa de desarrollo, permite el uso de ramas de trabajo para la experimentación y prueba de mecanismos, al igual que permite la revisión y regresión a versiones anteriores del programa por si ocurre cualquier inconveniente.

Como herramienta de planificación, se va a usar HacknPlan, debido a sus características y diseño enfocado al desarrollo de videojuegos y similitud con el estilo de Kanban, permitiendo crear y manejar tareas de una manera fácil y cómoda.

4.2. Metodologías de obtención de datos

Para la obtención de datos se van a realizar pruebas del juego con participantes principalmente de la universidad de Talca, en un rango de edad de entre 20 a 30 años, se buscará hacer estas pruebas con un grupo de entre 20 a 30 personas donde se les hará probar el juego y hacer pasar 4 niveles, con los primeros 2 siendo los niveles prediseñados y los últimos 2 serán los generados proceduralmente, eso sí, cuando pasen los primeros 2 niveles estos tendrán que rellenar una primera parte del cuestionario y cuando terminen los otros tendrán que rellenar una segunda parte del cuestionario que poseerá las mismas preguntas que la primera parte y algunas extra, esto para poder hacer una comparación entre su experiencia pasando niveles prediseñados y su experiencia con niveles generados proceduralmente, al igual que saber sobre su experiencia en general y obtener datos sobre su angustia, ansiedad o miedo que experimentaron durante la prueba del videojuego. Para obtener datos de ambas formas de creación de niveles, se buscará que se escoja de manera aleatoria a qué tipo de nivel se enfrentará el jugador y se colocará un código para poder identificar posteriormente qué tipo de mapa fue el que jugó cada participante.

El cuestionario que se va a realizar se usará como base los *Game Usability Heuristics (PLAY)* [26] propuestos por Heather Desurvire, para poder realizar preguntas principalmente cualitativas sobre la experiencia de los jugadores con respecto a la inmersión y del aprendizaje y dominio. Igualmente, se usará la herramienta *Subjective Units of Distress Scale (SUDS)* [27, 28], la cual, mediante una escala de 0 a 10, permite evaluar la angustia y la ansiedad que está experimentando un individuo, igualmente ambos cuestionarios están con sus respectivos links en el anexo A.4, incluyendo el cuestionario final, el cual será usado con los usuarios.

4.3. Carta Gantt

En la carta Gantt se describen las actividades realizadas durante este semestre y las funciones y requisitos para el producto final que se desarrollarán durante el primer semestre del próximo año; dentro de las actividades hechas este semestre están los informes entregados, y cada uno de estos tiene presente su duración total y la duración de sus requisitos por debajo. Cabe aclarar que el proyecto de videojuego se realizará en conjunto con Lisbeth González debido a que los dos tenemos temáticas similares, ambos queriendo hacer un juego de miedo; se trabajarán en conjunto partes del juego, pero cada uno encargándose de las cosas correspondientes a su investigación. También hay que aclarar que esto ya fue conversado con nuestro profesor guía. Para poder ver la carta Gantt diríjase al anexo A.1.

5. Implementación

5.1. Desarrollo de salas

La creación de las salas fue hecha en Blender, más específicamente en la versión 4.4.3.

Se buscó que las salas tuvieran diseños diferentes entre ellas para calzar con el look & feel del juego, combinando diferentes épocas y lugares de la humanidad. Igualmente, se hicieron 7 tipos de salas diferentes; 4 de ellas eran en forma de L, haciendo que se conectasen con 2 salas y tienen diferentes ángulos cada una, como se puede ver en la figura 5.1.

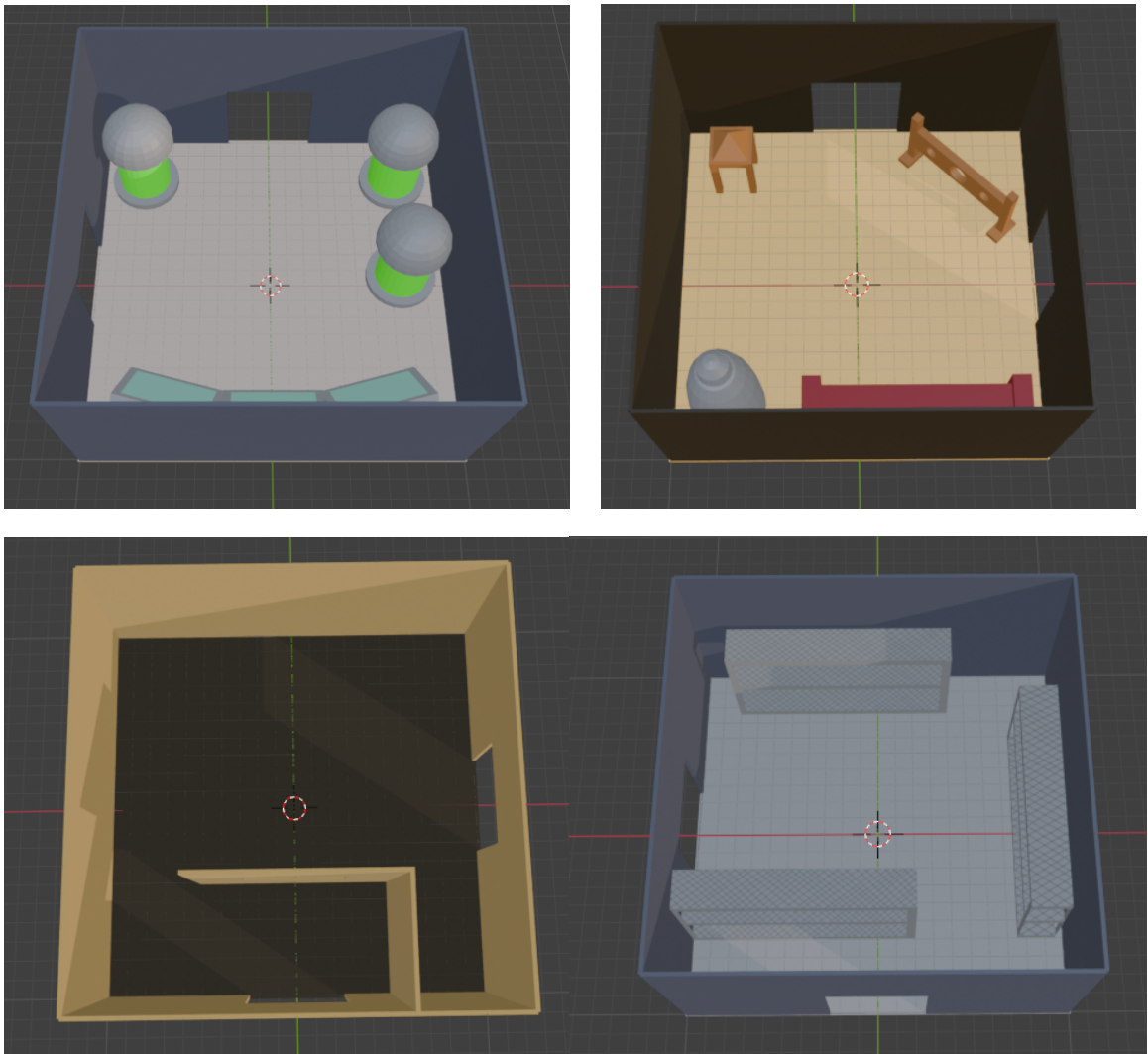
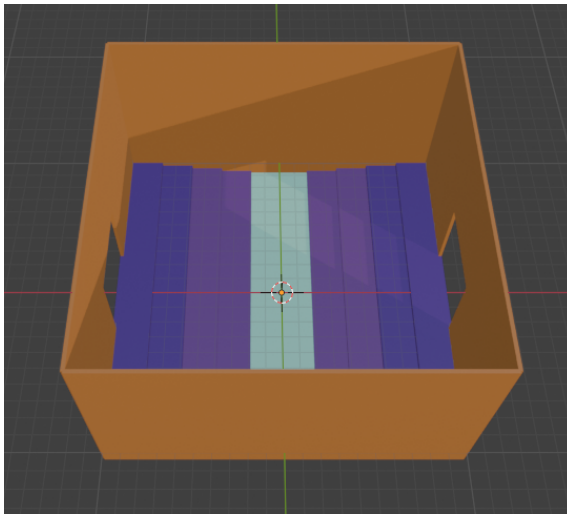
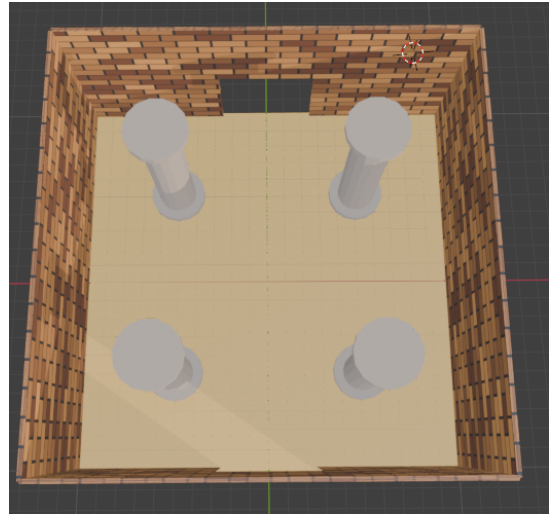


Figura 5.1: Salas en forma de L

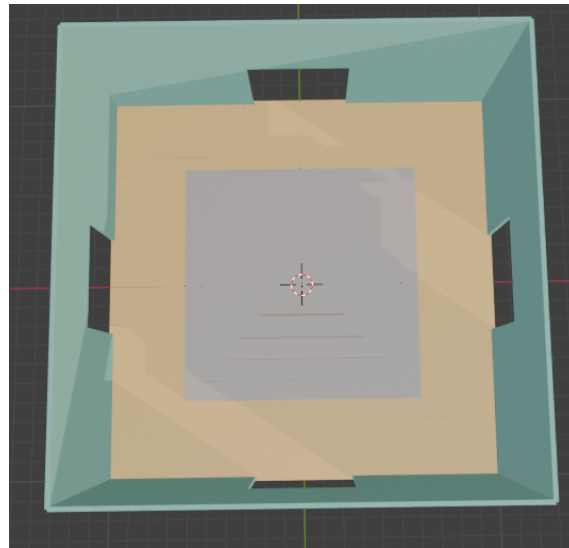
Igualmente, se crearon una sala con conexiones horizontales y otra con conexiones verticales y, por último, una sala que tuviera 4 conexiones, permitiendo que cualquier sala pudiera estar al lado de ella, como se puede ver en la figura 5.2.



(a) Sala Horizontal



(b) Sala Vertical



(c) Sala con 4 conexiones

Figura 5.2: Las otras Salas

5.2. Wave Function Collapse

Como se explicó en el marco teórico, Wave Function Collapse, o WFC, es un algoritmo de generación procedural, el cual genera contenido, o en este caso niveles, a través del “colapso” de posibilidades según reglas predefinidas, de esta forma garantizando un resultado más coherente.

Dentro del proyecto se creó un grupo de salas que se utilizaron tanto para el mapa prediseñado como para el mapa hecho utilizando WFC; para que el algoritmo pudiera usar las salas, se crearon 2 códigos, el primero siendo “WFCNode” (Listing 5.1). Este código permite generar “ScriptableObject” llamados nodos; a estos se les daba el prefab de una sala y sus vecinos para aplicar las restricciones:

```

1 [CreateAssetMenu (fileName = "WFCNode", menuName="WFC/Node")]
2 [System.Serializable]
3 public class WFCNode : ScriptableObject
4 {
5     public string Nombre;
6     public GameObject Prefab;
7     public WFC_Connection Arriba;
8     public WFC_Connection Abajo;
9     public WFC_Connection Izquierda;
10    public WFC_Connection Derecha;
11 }
12
13 [System.Serializable]
14 public class WFC_Connection
15 {
16     public List<WFCNode> NodosCompatibles = new List<WFCNode>();
17 }

```

Listing 5.1: Código WFCNode

Con este código como una base, después se creó el otro código principal, el “WFC-Builder”, el cual permitía la creación de los mapas utilizando la Wave Function Collapse. Este código posee una lista con todos los nodos, al igual que un alto y un ancho, los cuales se usan para crear una cuadrilla, al igual que una sala de inicio; esta sala de inicio es un WFCNode que hace que en cada nueva iteración del WFC esta sala siempre se mantenga igual. Esto se hizo para que siempre se pueda conectar de una forma u otra a la nave, que es donde los jugadores siempre tenían que regresar.

Este código empieza con la creación de la cuadrilla, donde se ubicarán los nodos, y empezará colocando la sala de inicio en una posición específica. Posteriormente, el código empieza a añadir a todos los vecinos de la sala de inicio que estén dentro de la cuadrilla para poder colapsarlos y para eso primero se busca si la sala está dentro de los nodos compatibles de los vecinos, y en aquellos que sí, se guardan esos vecinos y se eliminan los que no salgan como posible vecino, como se muestra en Listing 5.2 y en Listing 5.3.

```

1 private Vector2Int[] offsets = new Vector2Int[]
2 {
3     new Vector2Int(0, 1) //Arriba
4     new Vector2Int(0, -1) //Abajo

```

```
5     new Vector2Int(1, 0) //Derecha
6     new Vector2Int(-1, 0) //Izquierda
7 }
8 ...
9 public void CollapseWorld()
10 {
11     ...
12     for(int i = 0; i < offsets.Length; i++)
13     {
14         Vector2Int neighbour = new Vector2Int(x + offsets[i].x, y +
15         offsets[i].y);
16
17         if(IsInsideGrid(neighbour))
18         {
19             WFCNode neighbourNode = _grid[neighbour.x, neighbour.y];
20
21             if(neighbourNode != null )
22             {
23                 switch (i)
24                 {
25                     case 0:
26                         WhittleNodes(potentialNodes, neighbourNode.
27                         Abajo.NodosCompatibles);
28                         break;
29                     case 1:
30                         WhittleNodes(potentialNodes, neighbourNode.
31                         Arriba.NodosCompatibles);
32                         break;
33                     case 2:
34                         WhittleNodes(potentialNodes, neighbourNode.
35                         Izquierda.NodosCompatibles);
36                         break;
37                     case 3:
38                         WhittleNodes(potentialNodes, neighbourNode.
39                         Derecha.NodosCompatibles);
40                         break;
41                 }
42             }
43             else
44             {
45                 if(!_toCollapse.Contains(neighbour)) _toCollapse.Add
```

```
        (neighbour);  
41         }  
42     }  
43 }  
44 }
```

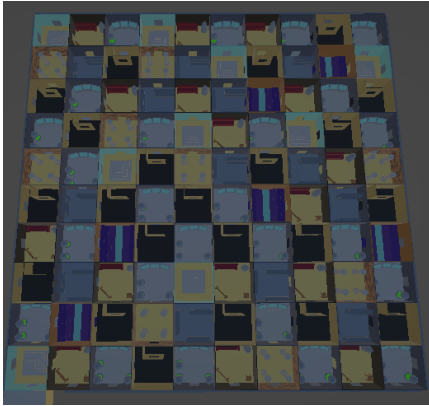
Listing 5.2: Código de colapso de vecinos

```
1 private void WhittleNodes(List<WFCNode> potentialNodes, List<WFCNode  
  > validNodes)  
2 {  
3     for(int i = potentialNodes.count - 1; i > -1; i--)  
4     {  
5         if(!validNodes.Contains(potentialNodes[i]))  
6         {  
7             potentialNodes.RemoveAt(i);  
8         }  
9     }  
10 }
```

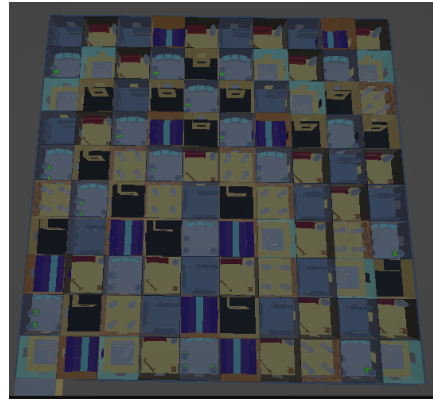
Listing 5.3: Código de vecinos validos

Estas funciones dentro del código se repiten hasta que ya no queden posibles vecinos y todos los nodos hayan sido colapsados, haciendo que haya correlación entre cada sala y sus vecinos, evitando que, por ejemplo, haya puertas que lleven a paredes, como se puede ver en la figura 6.1.

Adicionalmente, se adjunta un gameplay del juego que se puede ver en el anexo A.3.



(a) Mapa 1



(b) Mapa 2

Figura 5.3: Mapas generados proceduralmente por el algoritmo

6. Resultados

En esta sección se explicará como se hizo la “investigación”, al igual que los resultados conseguidos del cuestionario que rellenaron los jugadores participantes.

6.1. Aplicación del experimento

Para el desarrollo del experimento, se trabajó con estudiantes de la carrera de videojuegos, a los cuales se les explicó que serían parte de una memoria donde tendrían que jugar un prototipo de videojuego y responder un cuestionario para el uso de sus datos en dicha memoria, al igual que se les entregó un consentimiento informado donde se les explicaba que serían parte y donde podían firmar si querían serlo o no, como se puede ver en el anexo A.5. Tras eso se les entregó un pendrive con el prototipo de videojuego; para aquellos que tuvieran computador y quienes no, se les prestó un computador con el prototipo ya instalado. Igualmente, se les hizo entrega de un código QR, el cual tenía el link al cuestionario.

Cuando tenían todo listo, se les dieron las instrucciones del experimento, haciéndoles saber que primero tendrían que responder unas preguntas preliminares en el cuestionario para después jugar 2 niveles del juego y posteriormente responder preguntas relacionadas con su experiencia hasta el momento. Posteriormente, los jugadores iban a tener que seguir con los últimos 2 niveles del juego para responder las últimas preguntas, las cuales estaban relacionadas tanto con su experiencia con estos niveles generados proceduralmente, al igual que con su experiencia en general y opiniones finales.

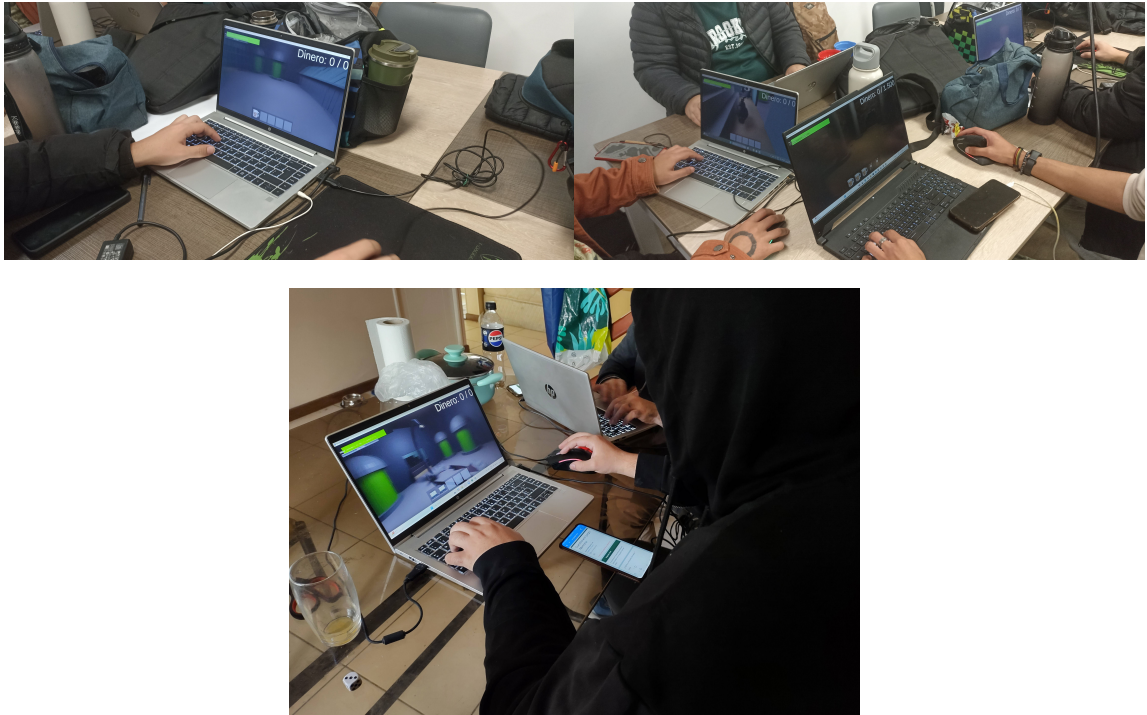


Figura 6.1: Pruebas con usuarios

6.2. Generales

¿Qué tan seguido juegas juegos de terror?
20 respuestas

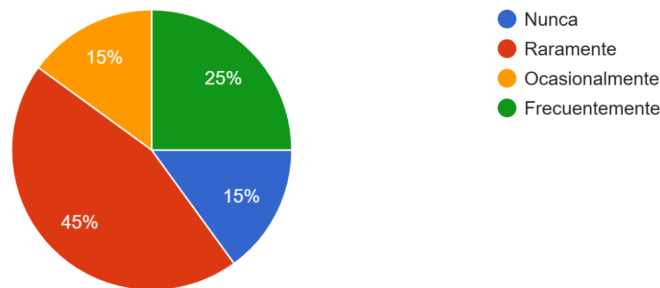


Figura 6.2: Cuan frecuente juegan juegos de terror

En general, la mayoría de los jugadores raramente han jugado un juego de terror antes de la investigación, como se puede ver en la figura 6.2. Esto es seguido por otro grupo de jugadores, un poco más pequeño que la mayoría, que dice que los juega con frecuencia, haciendo una división casi igualitaria entre gente que tiene experiencia con este tipo de juegos y gente que no. Cuando se les preguntó posteriormente el tipo de mapas que prefieren en un juego, no necesariamente de terror, hubo 9 jugadores que preferían mapas prediseñados debido al control que les otorgaba a los desarrolladores, permitiendo una mayor inmersión y atención a los detalles. En contraparte, 8 jugadores preferían juegos con mapas al azar debido a que les gusta más el desafío que estos otorgan; las otras 3 personas no tenían alguna preferencia, ya que depende del juego o en qué está enfocado.

6.3. Mapas prediseñados

¿Cuál es tu nivel de miedo?

20 respuestas

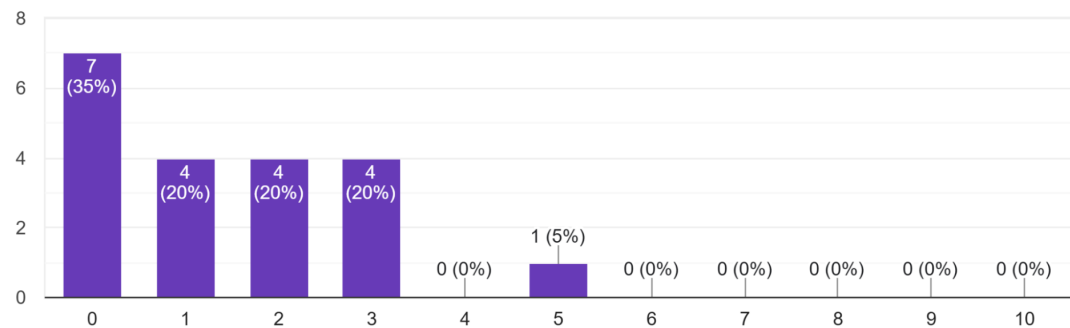


Figura 6.3: Nivel de miedo en mapas prediseñados

Como se puede ver en la figura 6.3, los jugadores experimentaron mayoritariamente niveles bajos de miedo con el juego, siendo “0” la opción más seleccionada. A pesar de esto, igual hubo 3 elementos principales que les causaron sustos a los jugadores en los niveles prediseñados; estos serían los sonidos del juego, el perderse en el mapa al experimentarlo por primera vez y los enemigos que se encontraban a través del nivel. Todas estas razones se mantienen bastante en nivel unas con las otras, no habiendo ninguna que sobresaliera por sobre otra.

Igualmente, se les preguntó si el diseño del nivel guio intencionalmente su miedo, a lo que la gran mayoría dice que el nivel no guio intencionalmente su miedo, aunque sí había ciertas cosas del nivel que sí generaban un ambiente de miedo, como lo sería la iluminación. La mayoría de los jugadores estaba de acuerdo en que faltaban detalles o conectividad entre las diferentes salas.

Otra de las preguntas que se hizo fue si es que el usuario se perdió dentro del nivel. Ante esto, la respuesta de los usuarios fue dividida entre gente que sí se perdió y

la gente que no. Como se puede ver en la figura 6.4, la mayoría de la gente que se perdió comentaba que la razón principal fue la repetición entre las salas, haciendo que fuera muy difícil orientarse. Tras volver a jugarlo, la gran mayoría encontró que el segundo nivel fue más fácil, debido a que entendían mejor qué cosas tenían que hacer, al igual que ahora conocían mejor la composición del nivel, haciendo que, a pesar de que en un inicio la gente se perdía, el nivel prediseñado terminara siendo bastante fácil, como se puede ver en la figura 6.5.

¿Qué tan intuitivo fue el diseño del nivel?

20 respuestas

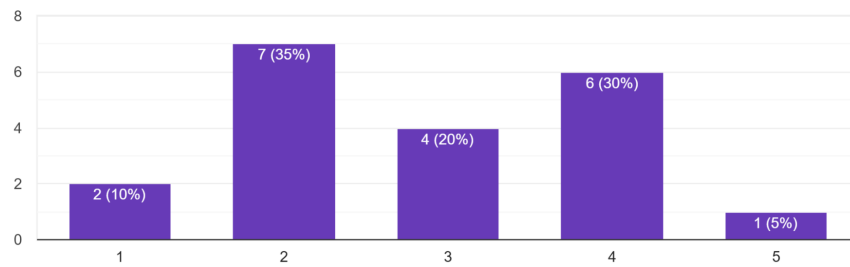


Figura 6.4: Qué tan intuitivos fueron los mapas prediseñados

¿Qué tan equilibrada estuvo la dificultad?
20 respuestas

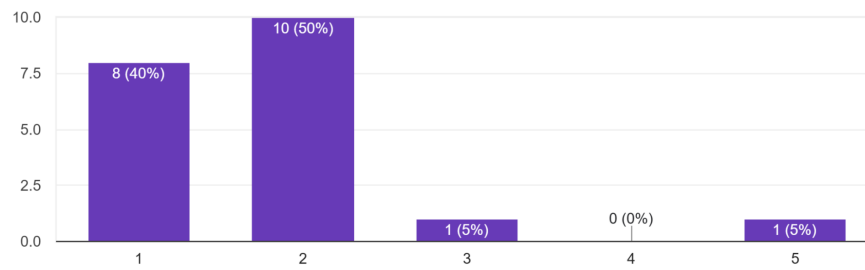


Figura 6.5: Qué tan equilibrados fueron los mapas prediseñados

6.4. Mapas generados proceduralmente

¿Cuál es tu nivel de miedo?

20 respuestas

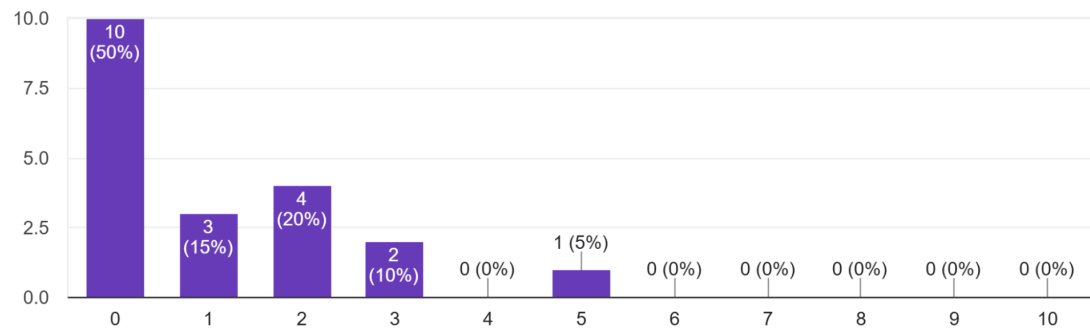


Figura 6.6: Nivel de miedo en mapas generados proceduralmente

Cuando se llega a los mapas generados proceduralmente, hubo menores niveles de miedo en comparación a los mapas prediseñados, como se puede ver en la figura 6.6. En esta ocasión, cuando se les preguntó qué fue lo que les asustó, la mayoría comentó que los enemigos, pero hubo un número de gente que se quejaba de los bugs que ocurrían.

Cuando se les preguntó si el nivel guió su miedo, la gran mayoría dijo que el nivel no guió su miedo y comentaban un error que ocurrió en el juego que provocaba que el nivel estuviera más iluminado de lo normal, al igual que hubo un número de personas a las que se les generó un nivel bastante corto y tuvieron que reiniciar el juego.

En contraste con los niveles en el mapa prediseñado, en este caso la gran mayoría comentó que sí se perdió en este nivel, como se puede ver en la figura 6.7. Igualmente, en esta pregunta algunos añadieron que empezaron a implementar estrategias para poder recordar el camino de vuelta, como lo sería el dejar ciertos objetos en el camino. Tras volver a jugarlo, el nivel se les hizo más fácil a las personas, aunque igual

seguía siendo más difícil que el mapa prediseñado, como se puede ver en la figura 6.8. Esto se debe a que ahora los jugadores ya sabían que el mapa era diferente, haciendo que las personas se empezaran a enfocar cada vez más en recordar el camino de vuelta, al igual que empezaron a hacer trayectos más pequeños y a empezar a buscar mejor por objetos en cada sala.

Cuando se les preguntó si se dieron cuenta sobre si los mapas eran generados proceduralmente, la mayoría comentó que sí se dio cuenta por la forma en que estaban colocadas las salas, incluso por si les tocaban niveles más cortos por errores. Cuando se les preguntó si los mapas generados proceduralmente hacían el juego más impredecible, la mayoría respondió que sí se volvía más impredecible, al igual que más injusto y regulable, como se muestra en la figura 6.9, aunque varios respondieron que la injusticia del nivel provenía de que no les aparecían los suficientes objetos para completar la meta o porque no se les generó un buen mapa en el cual jugar.

¿Qué tan intuitivo fue el diseño del nivel?
20 respuestas

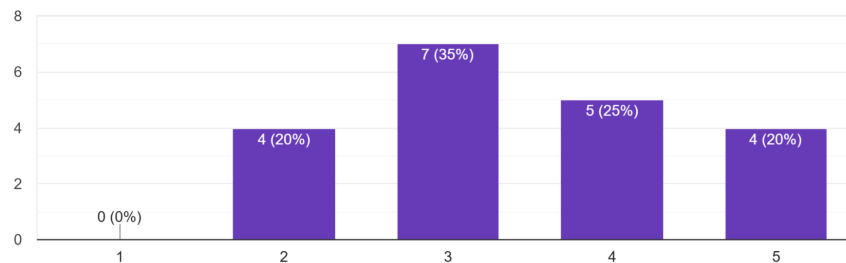


Figura 6.7: Qué tan intuitivos fueron los mapas generados proceduralmente

¿Qué tan equilibrada estuvo la dificultad?

20 respuestas

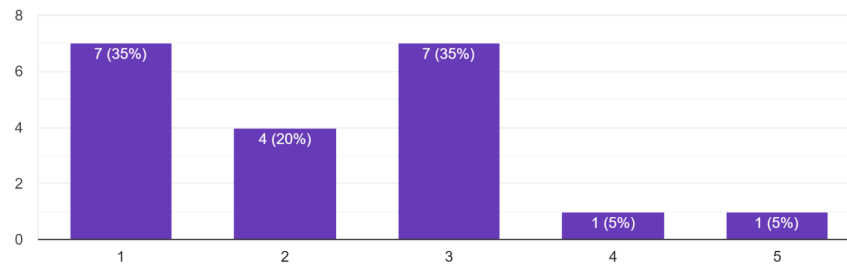


Figura 6.8: Qué tan equilibrados fueron los mapas generados proceduralmente

¿Los mapas creados al azar por el juego hicieron que el juego fuera más rejugable?

20 respuestas

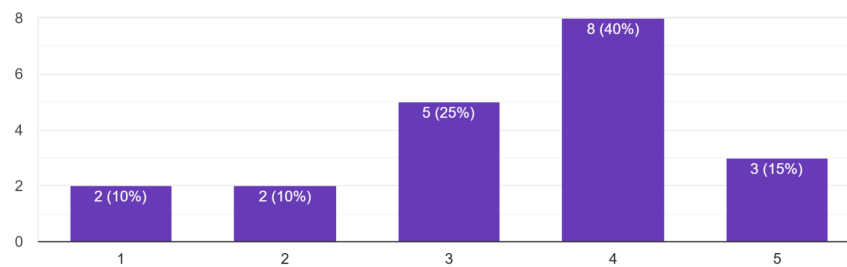


Figura 6.9: Qué tan rejugables hicieron los mapas generados proceduralmente al juego.

6.5. Apartado Final

Al final se les preguntó a los jugadores qué tipo de mapa les pareció más aterrador. Ante esto no hubo un tipo de mapa que sobresaliese, habiendo gente diciendo que los mapas prediseñados le produjeron más miedo y otros que mencionaban que los generados proceduralmente les daban más miedo, no habiendo un ganador; igualmente se les preguntó qué tipo de mapa preferían y, en general, la mayoría coincidía en que una mezcla era la mejor opción, queriendo que un juego pueda utilizar tanto mapas generados proceduralmente como mapas prediseñados, por los beneficios que ambos otorgan.

7. Conclusiones

En conclusión, la investigación se pudo realizar con éxito, pudiendo probar el prototipo de juego con un número deseado de personas, logrando entender y reforzar diferentes cosas relacionadas con la comparación entre mapas prediseñados vs. mapas generados proceduralmente; el mapa prediseñado a pesar de que no dio mucho miedo a los jugadores, varios de los que los preferían comentaban su interés debido al control que posee el desarrollador sobre el mapa, en contraparte, aunque cuando se les preguntó si el mapa generado proceduralmente les dio miedo, la mayoría respondió que tuvo menores niveles de miedo en comparación al mapa prediseñado, al final no hubo un mapa que fuera más aterrador que otro habiendo varios puntos a favor o en contra de ambos mapas, aunque lo que resalto más de los mapas generados proceduralmente fue su capacidad de rejugabilidad. A pesar de esto, la hipótesis no se cumplió, debido a que no se logró notar una mejora considerable en el miedo que sentían los jugadores en los mapas generados proceduralmente en comparación con el mapa prediseñado.

7.1. Trabajo futuro

Algunas recomendaciones que se pueden dar para una futura investigación serían que los jugadores jueguen por más niveles, esto para que logren notar con mayor facilidad la diferencia entre ambos estilos de mapas, pudiendo acomodarse más a los mapas prediseñados y también puedan experimentar mejor los mapas generados proceduralmente. Igualmente, poder modificar la posición de inicio en los mapas generados proceduralmente, para poder lograr encontrar la que genere menos problemas con la generación de mapas.

También se podrían mejorar otros aspectos del juego, como lo serían los elementos visuales o sonoros, debido a que estos afectan cómo los jugadores reaccionan ante el juego, pudiendo ser estos uno de los factores por el cual el prototipo no fue tan aterrador.

Igualmente, hay que tener en cuenta que estos siempre tendrían que ser iguales en ambos tipos de mapas, para que no exista ningún tipo de ventaja entre uno y otro a la hora de dar miedo.

Bibliografía

- [1] John Szczepaniak and Alex Wiltshire. *Japansoft: An Oral History*. Read-Only Memory, 2020.
- [2] Pratham Trikha, Deeksha Khosla, Ammu Ria, Tushar Dadwal, and Harjeet Kaur. A research on improvements in horror game development. In *KILBY 100 7th International Conference on Computing Sciences 2023*, 2023. <https://ssrn.com/abstract=4488186>.
- [3] Habgood, Jacob, Overmars, and Mark. *The Game Maker's Apprentice: Game Development for Beginners*, chapter Game Design: Levels and Features, pages 149–166. Apress, Berkeley, CA, 2006. https://doi.org/10.1007/978-1-4302-0159-5_8.
- [4] Miguel Frade, F. F. Vega, and C. Cotta. Automatic evolution of programs for procedural generation of terrains for video games. *Soft Computing*, 16:1893–1914, 2012. <http://dx.doi.org/10.1007/s00500-012-0863-z>.
- [5] Nathan Brewer. Computerized dungeons and randomly generated worlds: From rogue to minecraft [scanning our past]. *Proceedings of the IEEE*, 105(5):970–977, 2017.
- [6] Konstantinos Ntokos. Level of fear”: Analysis of fear spectrum into a tool to support horror game design for immersion and fear. *An International Journal (CGDEIJ)*, 1(33-43), 2018. <https://airccse.org/journal/ijcgde/papers/1113cgdeij03.pdf>.
- [7] Zeekerss. Lethal company on steam, oct 2023. https://store.steampowered.com/app/1966720/Lethal_Company. Último acceso: 22 de octubre de 2024.

- [8] thePetHen, Skog, Zorro, Wilnyl, and Philip. Content warning on steam, apr 2024. https://store.steampowered.com/app/2881650/Content_Warning. Último acceso: 22 de octubre de 2024.
- [9] Kinetic Games. Phasmophobia on steam, sep 2020. <https://store.steampowered.com/app/739630/Phasmophobia/>. Ultimo acceso: 22 de octubre de 2024.
- [10] Victoria Hawco. The ludic impact of horror games on the body: and. *Hungarian Journal of English and American Studies*, 29(2):381–400, 2023. <https://sciendo.com/article/10.30608/hjeas/2023/29/2/9>.
- [11] Kenneth Hullett. The science of level design. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, FDG '10, page 262–264, New York, NY, USA, 2010. Association for Computing Machinery. <https://doi.org/10.1145/1822348.1822387>.
- [12] Chongyang Ma, Nicholas Vining, Sylvain Lefebvre, and Alla Sheffer. Game level layout from design specification. In *Computer Graphics Forum*, volume 33, pages 95–104. Wiley Online Library, 2014.
- [13] Lenny Smith. Creating tension through game-centric design in survival horror video games. B.s. thesis, Tampere University of Applied Sciences, mar 2023. <http://www.theseus.fi/handle/10024/792646>.
- [14] Julio Ginter-Agreda. Liminal space: The transient experience of the architecturally obsolete, Apr 2024. <https://open.library.ubc.ca/collections/graduateresearch/42591/items/1.0442069>.
- [15] Saeed Amiri-Chimeh, Hassan Haghighi, and Mojtaba Vahidi-Asl. Jahan: A framework for procedural generation of game maps from design specifications. *Entertainment Computing*, 50:100644, 2024. <https://www.sciencedirect.com/science/article/abs/pii/S1875952124000120>.
- [16] Arunpreet Sandhu, Kyle Mitchell, and Joshua McCoy. Tileterror: A system for procedurally generating 2d horror maps. In *AIIDE Workshops*, 2021. <https://exagworkshop.institutedigitalgames.com/archive/sandhu2021tileterror.pdf>.

- [17] Hendrawan Armanto, Harits Ar Rosyid, et al. Evolutionary algorithm in game—a systematic review. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 2023. <https://kinetik.umm.ac.id/index.php/kinetik/article/view/1714>.
- [18] AM Connor, TJ Greig, and J Kruse. Evolutionary generation of game levels. *EAI Endorsed Transactions on Creative Technologies*, 2018. <https://openrepository.aut.ac.nz/items/2036dcdf-3ea9-4240-a3d4-3d07b71a3508>.
- [19] William L Raffe, Fabio Zambetta, and Xiaodong Li. A survey of procedural terrain generation techniques using evolutionary algorithms. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE, 2012. <https://ieeexplore.ieee.org/abstract/document/6256610>.
- [20] Leonardo Tortoro Pereira, Paulo Victor de Souza Prado, Rafael Miranda Lopes, and Claudio Fabiano Motta Toledo. Procedural generation of dungeons’ maps and locked-door missions through an evolutionary algorithm validated with players. *Expert Systems with Applications*, 180:115009, 2021. <https://www.sciencedirect.com/science/article/abs/pii/S0957417421004504>.
- [21] Darui Cheng, Honglei Han, and Guangzheng Fei. Automatic generation of game levels based on controllable wave function collapse algorithm. In *Entertainment Computing–ICEC 2020: 19th IFIP TC 14 International Conference, ICEC 2020, Xi’an, China, November 10–13, 2020, Proceedings 19*, pages 37–50. Springer, 2020. https://link.springer.com/chapter/10.1007/978-3-030-65736-9_3.
- [22] Arunpreet Sandhu, Zeyuan Chen, and Joshua McCoy. Enhancing wave function collapse with design-level constraints. In *Proceedings of the 14th International Conference on the Foundations of Digital Games, FDG ’19*, pages 1–9, New York, NY, USA, 2019. Association for Computing Machinery. <https://doi.org/10.1145/3337722.3337752>.
- [23] Howard Lei, Farnaz Ganjeizadeh, Pradeep Kumar Jayachandran, and Pinar Ozcan. A statistical analysis of the effects of scrum and kanban on software development projects. *Robotics and Computer-Integrated Manufacturing*, 43:59–67, 2017.

- [24] Krunal Bhavsar, Vrutik Shah, and Samir Gopalan. Scrumban: An agile integration of scrum and kanban in software engineering. *International Journal of Innovative Technology and Exploring Engineering*, 9(4):1626–1634, 2020. <https://www.ijitee.org/portfolio-item/D1566029420/>.
- [25] Necmettin Ozkan, Sevval Bal, Tugba Gurgun Erdogan, and Mehmet Şahin Gök. Scrum, kanban or a mix of both? a systematic literature review. In *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*, pages 883–893. IEEE, 2022.
- [26] Heather Desurvire and Charlotte Wiberg. Game usability heuristics (play) for evaluating and designing better games: The next iteration. In A. Ant Ozok and Panayiotis Zaphiris, editors, *Online Communities and Social Computing*, pages 557–566, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. https://link.springer.com/chapter/10.1007/978-3-642-02774-1_60#citeas.
- [27] Joseph Wolpe. Subjective units of distress scale. *Journal of EMDR Practice and Research*, 1969.
- [28] Barry A Tanner. Validity of global physical and emotional suds. *Applied psychophysiology and biofeedback*, 37:31–34, 2012.

A. Anexos

A.1. Carta Gantt

Debido al tamaño de la carta Gantt, se decidió colocar un enlace para hacer más fácil su lectura:

<https://docs.google.com/spreadsheets/d/1obu838jnuTnbJUmkkuXW3CITG7GA1NVBLr8xDPeedit?usp=sharing>

A.2. GDD

Aquí dejo el enlace al GDD:

https://docs.google.com/document/d/13U_h4426rJDNYzTu2MTK1NQki_aXrmZEyU2lK4qebDPeedit?usp=sharing

A.3. Video Gameplay

Video con gameplay del prototipo de juego:

<https://youtu.be/HyBR4G46zgU>

A.4. Cuestionarios

Game Usability Heuristics (PLAY): https://link.springer.com/chapter/10.1007/978-3-642-02774-1_60

Subjective Units of Distress Scale (SUDS): <https://psycnet.apa.org/doiLanding?doi=10.1037%2F05183-000>

Cuestionario usado en el experimento:

Cuestionario Memoria

¡Gracias por participar!, este cuestionario se enfoca en tu experiencia jugando Project Omega. Tus respuestas serán anónimas y no tomará más de unos minutos. No dudes en desarrollar tus ideas; tus comentarios son valiosos.

** Indica que la pregunta es obligatoria*

Antes de empezar

Un par de preguntas introductorias.

1. ¿Qué tan seguido juegas juegos de terror? *

Marca solo un óvalo.

- ☐ Nunca
- ☐ Raramente
- ☐ Ocasionalmente
- ☐ Frecuentemente

2. ¿Prefieres mapas creados a mano o mapas creados al azar por el juego? ¿Por qué? *

Primeros niveles

Tras pasar los primeros 2 niveles, responder estas preguntas con sinceridad.

3. ¿Cuál es tu nivel de miedo? *

Marca solo un óvalo.

	0	1	2	3	4	5	6	7	8	9	10	
Nad	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	Mucho miedo

4. ¿Qué cosa te asustó más? *

5. ¿Sentiste que el diseño del nivel guio intencionalmente tu miedo? ¿Por qué? *

6. ¿Qué tan intuitivo fue el diseño del nivel? *

Marca solo un óvalo.

	1	2	3	4	5	
Basí	☐	☐	☐	☐	☐	Nada intuitivo

7. ¿Te perdiste? Si es así, ¿dónde o cómo? *

8. ¿Qué tan equilibrada estuvo la dificultad? *

Marca solo un óvalo.

1	2	3	4	5		
<hr/>						
Muy	☐	☐	☐	☐	☐	Muy difícil
<hr/>						

9. Tras volver a jugarlo, ¿te pareció más fácil el nivel? ¿Por qué? *

Últimos niveles

Tras pasar los últimos 2 niveles, responder estas preguntas repetidas al igual que unas extras

10. ¿Cuál es tu nivel de miedo? *

Marca solo un óvalo.

	0	1	2	3	4	5	6	7	8	9	10	
Nad	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	Mucho miedo

11. ¿Qué cosa te asustó más? *

12. ¿Sentiste que el diseño del nivel guio intencionalmente tu miedo? ¿Por qué? *

13. ¿Qué tan intuitivo fue el diseño del nivel? *

Marca solo un óvalo.

	1	2	3	4	5	
Bast	☐	☐	☐	☐	☐	Nada intuitivo

14. ¿Te perdiste? Si es así, ¿dónde o cómo? *

15. ¿Qué tan equilibrada estuvo la dificultad? *

Marca solo un óvalo.

1 2 3 4 5

Muy ☐ ☐ ☐ ☐ ☐ Muy difícil

16. Tras volver a jugarlo, ¿te pareció más fácil el nivel? ¿Por qué? *

17. ¿Te diste cuenta de que los niveles eran aleatorios? ¿Cómo? *

18. ¿Los mapas creados al azar por el juego hicieron que el juego fuera más impredecible o aterrador? *

19. ¿Los mapas creados al azar por el juego hicieron que el juego fuera más rejugable? *

Marca solo un óvalo.

1 2 3 4 5

Total ☐ ☐ ☐ ☐ ☐ Totalmente de acuerdo

20. ¿Los mapas creados al azar por el juego se sintieron menos o más justos que los hechos a mano? ¿Por qué? *

Comparación final

Últimas preguntas sobre la experiencia completa.

21. ¿Qué tipo de nivel (a mano / creados al azar por el juego) te pareció más aterrador? ¿Por qué? *

22. ¿Preferirías un juego con mapas hechos solo a mano, solo creados al azar por el juego o una mezcla? ¿Por qué? *

Google no creó ni aprobó este contenido.

Google Formularios

A.5. Consentimiento Informado

Consentimiento Informado:



CONSENTIMIENTO INFORMADO

VÁLIDO DESDE: 27 / 05 / 2025

HASTA: 31 / 12 / 2025

Título del Proyecto: Comparación entre mapas Generados Proceduralmente y mapas con Diseño Predefinido para juegos de terror

Financiado por: Diego Valderrama

Patrocinante: _____

Estimado Sr. (Sra.):

El propósito de este documento es entregarle toda la información necesaria para que usted pueda **DECIDIR LIBREMENTE SI DESEA PARTICIPAR en la INVESTIGACIÓN** que se le ha explicado verbalmente, y que a continuación se describe en forma resumida.

DESCRIPCIÓN DEL ESTUDIO Y OBJETIVO(S)

Este estudio tiene como objetivo observar y analizar las reacciones emocionales de los jugadores al experimentar un prototipo de videojuego de terror desarrollado como parte de una memoria universitaria. La información obtenida podría aportar al diseño y desarrollo en futuros videojuegos de terror.

BENEFICIOS

Los beneficios indirectos son como los resultados del estudio pueden contribuir a mejorar la calidad de diseño en videojuegos de terror.

PROCEDIMIENTOS

Participarán entre 20 y 35 voluntarios/as mayores de edad que sean estudiantes de la carrera de videojuegos. El estudio se realizará en una sala equipada con computadoras, donde el participante jugará una sesión de entre 15 a 25 minutos en la cual entre medio se le hará responder partes de una encuesta de evaluación.

RIESGOS Y MOLESTIAS

Podría experimentar sensaciones de miedo, ansiedad moderada o sobresaltos debido al contenido del videojuego.

COSTOS

La participación no implica ningún costo económico para el/la participante.



COMPENSACIONES

No se entregará ningún tipo de compensación.

CONFIDENCIALIDAD

Todos los datos recolectados serán tratados con confidencialidad. Se usará un Google Forms donde no se recopilará ningún dato personal de los participantes, igualmente este mismo y un documento Excel será donde se almacenarán los datos hasta finales del año 2025 donde solamente el investigador tendrá acceso.

DESTINO DEL MATERIAL OBTENIDO

Una vez concluido el estudio, los datos serán analizados y utilizados exclusivamente en el informe de memoria. Luego de la finalización del año 2025 toda la información será eliminada permanentemente.

COMUNICACIÓN DE RESULTADOS

Los resultados generales del estudio serán descritos en la memoria del investigador, donde las/los participantes podrán acceder posterior a la publicación de la memoria.

COMUNICACIÓN CON EL INVESTIGADOR Y CEC

En caso de dudas, puede contactar al investigador responsable:

Nombre: Diego Valderrama

Correo: dvalderrama20@alumnos.utalca.cl

Teléfono: +56940170380

Horario: Lunes a viernes, 10:00 a 17:00 hrs.

También puede comunicarse con el Comité Ético Científico de la Universidad de Talca (cec@utalca.cl, Teléfono 71-2203065, lunes a jueves: de 09:30-13:00 y de 14:30-16:00 hrs).

Estimado/a participante recuerde que la decisión de colaborar es absolutamente suya. Puede aceptar o rechazar participar en la investigación, e incluso arrepentirse de su primera decisión en el momento que usted estime conveniente para lo cual puede contactarse con Diego Valderrama, Universidad de Talca, lunes a viernes, 10:00 a 17:00 hrs. para firmar la **hoja de revocación**.

Desde ya muchas gracias, saluda cordialmente a usted.

Diego Valderrama



DECLARACIONES

1. He recibido una explicación satisfactoria sobre el propósito de la investigación la cual consiste en observar y analizar las reacciones emocionales que se producen al experimentar un prototipo de videojuego de terror. Este estudio busca aportar conocimientos al diseño y desarrollo de futuros desarrollos de videojuegos.
2. He sido informado/a sobre las eventuales molestias, incomodidades y riesgos de mi participación en la investigación.
3. He sido también informado/a que los procedimientos que se realicen no implican un costo que yo deba asumir, es decir, mi participación en el procedimiento no involucra un costo económico alguno para mí.
4. Estoy en pleno conocimiento de que la información obtenida será manejada de manera absolutamente confidencial, esto significa que sólo el equipo investigador tendrá acceso a mis datos y nadie más. En caso de que la información obtenida del estudio sea publicada ésta se mantendrá anónima, ello significa que no aparecerá ningún dato con el que puedan identificarme en libros, revistas y otros medios de publicidad derivadas de la investigación ya descrita.
5. Sé que la **DECISIÓN** de participar en esta investigación, es **ABSOLUTAMENTE VOLUNTARIA**. Si no deseo participar en ella, o una vez iniciada la investigación no deseo seguir colaborando, puedo hacerlo sin tener que dar ninguna explicación. Para esto último solo debo presentarme la Universidad de Talca, lunes a viernes de 10:00 a 17:00 hrs, con el Sr., Sra., Diego Valderrama para firmar la hoja de revocación.
6. Adicionalmente, los/las investigadores/as responsable/s Diego Valderrama (dvalderrama20@alumnos.utalca.cl, +56940170380, lunes a viernes de 10:00 a 17:00 hrs.) han manifestado su voluntad de aclarar cualquier duda que me surja, antes, durante y después de mi participación en la actividad.
7. También puedo contactarme con el **Comité Ético Científico** de la Universidad de Talca (cec@utalca.cl, Teléfono 71-2-203065. Lunes a jueves: de 09:30-13:00 y de 14:30-16:00).



ACEPTACIÓN

He leído el documento, las declaraciones contienen una explicación satisfactoria sobre mi participación en la investigación y sobre la necesidad de hacer constar mi consentimiento, para lo cual firmo libre y voluntariamente, recibiendo en el acto copia de éste documento ya firmado.

Yo, _____,
Cédula de Identidad o Pasaporte N° _____ de nacionalidad
_____, mayor de edad, con domicilio
en _____
_____, **ACEPTO** participar en la investigación
titulada: "_____", y **AUTORIZO** al
señor(a) _____ investigador(a)
responsable del proyecto, y/o a quienes éste(a) designe como sus colaboradores directos
y cuya identidad consta al pie del presente documento, para realizar el(los) procedimiento(s)
requerido(s) por el proyecto de investigación descrito.

Fecha: ____ de _____ 20____ Hora: ____: ____

Firma de la persona que consiente: _____

Investigador/a responsable: _____
Nombre Firma

Co-investigador/a 1 : _____
Nombre Firma

Co-investigador/a 2 : _____
Nombre Firma



RECHAZO

He leído el documento y las declaraciones contienen una explicación satisfactoria sobre la investigación. Sin embargo, rechazo otorgar mi consentimiento, para cual firmo libre y voluntariamente el siguiente documento, recibiendo en el acto copia de éste ya firmado.

Yo, _____,
Cédula de Identidad o Pasaporte N° _____ de nacionalidad
_____, mayor de edad, con domicilio
en _____,
_____, **RECHAZO** participar en la investigación
titulada: "_____", y **NO AUTORIZO** al
señor(a) _____ investigador(a)
responsable del proyecto, y/o a quienes éste(a) designe como sus colaboradores directos
y cuya identidad consta al pie del presente documento, para realizar el(los) procedimiento(s)
requerido(s) por el proyecto de investigación descrito.

Fecha: ____ de _____ 20____ Hora: ____: ____

Firma de la persona que consiente: _____

Investigador/a responsable: _____
Nombre Firma

Co-investigador/a 1 : _____
Nombre Firma

Co-investigador/a 2 : _____
Nombre Firma



REVOCACIÓN

Mediante la presente revoco lo anteriormente firmado, para lo cual firmo este nuevo documento libre y voluntariamente, recibiendo en el acto copia de este documento ya firmado.

Yo, _____,
Cédula de Identidad o Pasaporte N° _____ de nacionalidad
_____, mayor de edad, con domicilio
en _____
_____, **REVOCO** lo anteriormente firmado.

Fecha: ____ de _____ 20____ Hora: ____: ____

Firma de la persona que consiente: _____

Investigador/a responsable: _____
Nombre Firma

Co-investigador/a 1 : _____
Nombre Firma

Co-investigador/a 2 : _____
Nombre Firma