



**Universidad**  
Zaragoza

## Trabajo Fin de Grado

Desarrollo de un videojuego comercial: "Descenso".

A commercial videogame's development: "Descenso".

Autor

David Orera Hernández

Director

Francisco José Serón Arbeloa

Escuela de Ingeniería y Arquitectura  
2016



# RESUMEN

## Desarrollo de un videojuego comercial: "Descenso".

La industria de los videojuegos, pese a su relativa juventud, ha crecido a un ritmo vertiginoso en los últimos años. En este mercado algunos grandes estudios han conseguido afianzarse y mueven la mayor parte del dinero con sus grandes proyectos o "triple A", productos que involucran a equipos multidisciplinares de titánicas dimensiones y presupuestos desorbitados.

Paralela al fenómeno anterior ha surgido una corriente de desarrolladoras *indies* o independientes. Pequeños equipos que, aprovechando la facilidad que otorgan las nuevas tecnologías, realizan proyectos de carácter más modesto, íntimo y original.

Este proyecto propone el desarrollo de "Descenso", un videojuego de corte independiente, y su posterior publicación y distribución mediante una campaña de mecenazgo. Para ello se han analizado y diseñado las principales características que ha de contener el juego y se ha implementado una demo que será publicada para favorecer dicha campaña de financiación.

La demo implementa tres algoritmos de generación de entornos: paso borracho, generación de laberintos de Prim y autómatas celulares. Además, contiene tres enemigos con comportamientos distintos y un sistema de armas y hechizos. En ella el jugador puede elegir entre dos entornos de juego que se generan de forma distinta en cada ejecución.

Esta demostración ha sido probada en dos sesiones con usuarios. En la primera, se ha validado el control del personaje y la generación de entornos. En la segunda, se ha medido el grado de satisfacción y desafío mediante una encuesta tras superar un nivel de juego. De esta segunda prueba se ha obtenido, además, información sobre los aspectos que los usuarios creen que deben ser potenciados o cambiados.





## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D<sup>a</sup>. \_\_\_\_\_,

con nº de DNI \_\_\_\_\_ en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)  
\_\_\_\_\_, (Título del Trabajo)

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, \_\_\_\_\_

Fdo: \_\_\_\_\_



## Contenidos

1. Introducción.....	1
2. Objetivos .....	2
3. Diseño del juego .....	3
3.1. Argumento .....	3
3.2. Jugabilidad.....	3
3.3. Mapas .....	4
3.4. Elementos .....	5
3.5. Cordura .....	5
3.6. Controles.....	5
4. Especificación.....	7
4.1. Tabla de requisitos .....	7
4.2. Diagrama de clases UML simplificado .....	9
5. Generación de entornos .....	12
5.1. Paso aleatorio en dos dimensiones modificado o paso borracho. ....	12
5.2. Generación de laberintos .....	14
5.3. Autómatas celulares .....	16
6. Enemigos .....	18
7. Herramientas.....	19
7.1. Gimp .....	19
7.2. Spriter Pro.....	19
7.3. Unity .....	19
8. Resultados .....	20
9. Validación y comunicación con usuarios .....	23
9.1. Formulario .....	23
9.2. Grupo de prueba.....	23
9.3. Pruebas presenciales.....	23
10. Conclusiones.....	25
11. Trabajo futuro .....	27
11.1. Campaña de financiación colectiva y distribución digital .....	27
11.2. Completar el videojuego.....	27
12. Diagrama temporal .....	29
13. Bibliografía.....	31
Anexo A. GPD.....	32

Anexo B. Formulario online .....	36
Anexo C. Sesión de pruebas 2 .....	41
Anexo D. Diagramas de estado de enemigos .....	48

## 1. Introducción

La industria de los videojuegos, pese a su relativa juventud, ha crecido a un ritmo vertiginoso en los últimos años. En este mercado algunos grandes estudios han conseguido afianzarse y mueven la mayor parte del dinero con sus grandes proyectos o "triple A", productos que involucran a equipos multidisciplinares de titánicas dimensiones y presupuestos desorbitados.

Paralela al fenómeno anterior ha surgido una corriente de desarrolladoras *indies* o independientes. Pequeños equipos que, aprovechando la facilidad que otorgan las nuevas tecnologías, realizan proyectos de carácter más modesto, íntimo y original. En muchas ocasiones aprovechan el vacío dejado por los juegos de corte generalista desarrollados por grandes compañías para ofrecer productos más específicos en cuanto a género, ambientación o dificultad. Su éxito radica en que a pesar de dirigirse a públicos más reducidos trabajan con inversiones inferiores y necesitan muchas menos ventas para obtener beneficios.

Este proyecto pretende sumarse a la segunda corriente desarrollando un videojuego que beba de la mezcla de dos géneros fácilmente combinables: plataformas de exploración o *metroidvania* y mazmorras de generación procedural o *roguelike*. Además, cuenta con una ambientación, como son los Mitos de Lovecraft, con una fiel comunidad detrás.

## 2. Objetivos

Siguiendo con la idea anteriormente expuesta, se propone desarrollar un videojuego de plataformas procedural en dos dimensiones ambientado en los Mitos de Lovecraft. El interés del proyecto radica en la generación algorítmica de los entornos y la caracterización y modelado del comportamiento de los antagonistas. Se pretende, además, ofrecer una experiencia con un grado de dificultad más exigente al de los juegos de plataformas actuales de las grandes compañías que tienden a la casualización e infantilización del género.

Tras su diseño, se pretende realizar una demo que muestre las características principales del juego y ponerla a prueba con jugadores pertenecientes al público objetivo en dos ocasiones. La primera prueba pretende verificar la generación de escenarios y tantear la respuesta de los usuarios a los controles y lo intuitivos que estos puedan resultar. El cometido de la segunda prueba es triple: comprobar la reacción de los jugadores ante el desafío que supone completar un nivel ya poblado con enemigos y otros elementos, medir su grado de satisfacción o frustración y detectar factores que deban ser mejorados o modificados.

Por último, ya al margen de este trabajo fin de grado, se pretende realizar una campaña de financiación colectiva mediante la cual se puedan conseguir los fondos necesarios para la publicación y distribución el juego. Esta campaña se apoyará en la distribución de la demo antes mencionada y en una campaña de publicidad e interacción con los usuarios en las redes sociales y comunidades relacionadas.

Las tareas a realizar para conseguir estos objetivos coinciden con los siguientes puntos de la estructura de este TFG:

3. Diseño del juego (Realizado por el firmante de este TFG)
4. Especificación (Realizada por el firmante de este TFG)
5. Generación de entornos (Algoritmos codificados por el firmante de este TFG)
6. Desarrollo de enemigos (Realizado por el firmante de este TFG)
9. Validación y comunicación con usuarios (Realizada por el firmante de este TFG)

Para generar la demo se ha usado como plataforma de desarrollo y motor Unity.

### 3. Diseño del juego

*"Que no está muerto lo que puede yacer eternamente y en los eones venideros hasta la muerte puede morir"<sup>[1]</sup>*

*"Meses rebuscando en viejos archivos, persiguiendo a ariscos eruditos avaros de conocimientos prohibidos. He perdido mi trabajo, mis amigos, y dicen que mi cordura en la búsqueda insana de aquello que me la pueda devolver.*

*Mi formación académica me impediría creer en todo lo que ahora persigo, si no fuese porque he visto... porque sé...*

*Estaba a punto de rendirme cuando he encontrado lo que parece una lista inocua y un extraño diagrama. Pero estos elementos, en las proporciones adecuadas, dispuestos según dicta el diagrama, me pondrán en contacto, por fin, con el ente capaz de devolverme aquello que mi alma anhela. Cualquier precio, cualquier sacrificio, me parecerá justo si puedo contemplarla una vez más."*

#### 3.1. Argumento

El universo de "Descenso" se rige por una serie de misterios, leyes y seres que el ser humano no es capaz de comprender, apenas le es posible vislumbrar su sombra. Sin embargo, unos pocos están dispuestos a lo que sea para doblar este poder a su voluntad, son los sectarios, individuos que renuncian poco a poco a su cordura a cambio de profundizar en los oscuros misterios del universo en busca de aquello que anhelan.

A lo largo del juego, el jugador va a acompañar a uno de estos sectarios en un viaje, a través de distintas localizaciones en el tiempo y el espacio, en el que intentará burlar a los poderes superiores en beneficio propio. Se enfrentará a criaturas que nunca deberían haber existido, encontrará hechizos y artefactos de poder y progresivamente irá cayendo en la locura hasta desencadenar el final de su viaje.

A pesar del trasfondo común de una búsqueda egoísta de poder, en cada partida de "Descenso" se va a generar un argumento distinto compuesto por una meta, es decir, para qué se busca ese poder, y por un camino compuesto por los obstáculos que hay que superar para obtenerlo. Este argumento, que influye en la generación de los niveles y su concatenación, se obtiene en parte aleatoriamente y en parte gracias a las respuestas que el jugador da a un test antes de cada partida.

#### 3.2. Jugabilidad

Una partida de "Descenso" consiste en responder a un sencillo test y superar cuatro niveles consecutivos. Cada uno de ellos tiene una serie de parámetros asignados a partir de las respuestas del jugador al test inicial. Estos parámetros son:

##### **-Ambiente**

El ecosistema en el que se integra el nivel. Define el tipo de construcciones y materiales que pueden aparecer, los objetos que se pueden recolectar, los enemigos que se pueden encontrar y todo el aspecto visual del entorno generado.

##### **-Misión**

El objetivo a cumplir antes de poder enfrentarse al jefe del nivel. Este podrá ser reunir una serie de objetos, derrotar a un enemigo concreto o simplemente encontrar la salida.

### **-Jefe**

Un enemigo al que hay que superar para poder completar el nivel. Aunque lo normal es derrotarles en combate, hay jefes que pueden ser superados de otras formas que el jugador debe descubrir.

Cada uno de estos niveles está compuesto por una serie de entornos interconectados. Para pasar del actual a uno de sus adyacentes el jugador debe cumplir la condición de salida del entorno. Esta condición puede ser acabar con todos los enemigos presentes o activar un interruptor que abra las puertas.

Según avance en su aventura el jugador encontrará armas y hechizos con los que hacer frente a los peligros y desbloqueará contenido para futuras partidas.

### **3.3. Mapas**

El mapa de un nivel está compuesto por una matriz de entornos interconectados y dividido en distintas regiones según sean laberintos, exteriores, cuevas o mazmorras. Cada una de estas regiones, en distintos colores en las figuras 1 y 2, tiene un tamaño variable dependiendo del nivel actual. Además, el tipo, cantidad y distribución de las mismas depende del ambiente al que pertenezca dicho mapa.

A continuación, se muestran dos ejemplos de mapas que coinciden con los implementados para la prueba con usuarios:

#### **Mansión**

En la mansión hay un jardín delantero y quizá un jardín trasero formados por entornos exteriores ocupando la intersección de la fila inferior con las primeras y últimas columnas de la matriz. Entre ambos jardines se generan laberintos, rellenando tanto en altura como en anchura, que componen el interior de la mansión.

x	x	Laberinto	Laberinto	x
x	x	Laberinto	Laberinto	x
Exterior	Exterior	Laberinto	Laberinto	Interior

*Figura 1. Distribución de entornos en el mapa de la mansión.*

#### **Templo**

El templo, por su parte, consta de un desierto, unas catacumbas y el propio interior. Las catacumbas están compuestas por cuevas en las filas inferiores de la matriz. Sobre las catacumbas se encuentran el desierto a la izquierda, entornos exteriores, y el interior del templo a la derecha, laberintos.

x	x	Laberinto	Laberinto	Laberinto
Exterior	Exterior	Laberinto	Laberinto	Laberinto
Cueva	Cueva	Cueva	Cueva	Cueva
Cueva	Cueva	Cueva	Cueva	Cueva

*Figura2. Distribución de entornos en el mapa del templo.*

### 3.4. Elementos

A lo largo de su aventura el jugador puede encontrar, entre otros, los siguientes tipos de elementos:

#### **-Armas**

Cada arma tiene una serie de atributos que la diferencian de las demás y pertenece a una categoría que define el tipo de ataque que puede realizar, si es a distancia o cuerpo a cuerpo. El personaje solo puede llevar un arma en cada momento.

#### **-Hechizos**

Los hechizos ofrecen posibilidades que afectan sensiblemente a la jugabilidad. Cada hechizo puede ser equipado como hechizo primario o como hechizo secundario y ofrece efectos distintos dependiendo de esta posición. El efecto primario es inmediato y puede ser lanzado por el jugador a voluntad, suponiéndole un coste de algún tipo: salud, cordura u otros. Por otro lado el efecto secundario otorga un estado que afectará continuamente al personaje mientras lo lleve equipado. El jugador solo puede llevar un hechizo en cada posición.

#### **-Consumibles**

Pociones y otros objetos que el personaje puede usar una única vez, produciéndole algún tipo de efecto inmediato o estado temporal o permanente.

#### **-Objetos de misión**

Consumibles especiales cuya función es, una vez recogidos todos, permitir el paso al siguiente nivel si formaban parte de la misión actual.

### 3.5. Cordura

La locura, y su caída en ella a través del terror, es un tema común en toda la obra de H.P. Lovecraft. Para ser fiel a la ambientación, el juego implementa un sistema de cordura en el que el jugador pierde puntos según avanza o toma ciertas decisiones. Este nivel de cordura influye en varias mecánicas y se ve reflejado de forma visual en el juego.

La pérdida de cordura es una mecánica más del juego y no ha de verse, a priori, como algo negativo. Pese a que conlleve efectos negativos como estados perjudiciales también potencia otros aspectos, como puede ser la potencia de algunos hechizos o el cambio de actitud de algunos enemigos; incluso puede permitir acceder a armas o hechizos especiales.

### 3.6. Controles

Se pretende que el control sea lo más sencillo e intuitivo posible, por lo que el jugador dispone de una limitada lista de acciones:

**-Saltar:** el personaje salta al pulsar el botón de salto y la duración del salto depende del tiempo que se mantenga pulsado el mismo.

**-Atacar:** el personaje ataca con el arma equipada al pulsar el botón de interacción, si no hay otra interacción posible.

**-Interactuar con el entorno:** al pulsar el botón de interacción el personaje interactúa con los elementos del entorno que lo permitan: puertas, botones, pociones, etc.

**-Equipar:** el personaje se equipa un objeto equipable al pulsar el botón de interacción tras situarse junto a él.

**-Lanzar hechizo:** si el personaje lleva equipado un hechizo puede utilizarlo pulsando el botón de lanzamiento de hechizo.

**-Equipar un hechizo como secundario:** si el personaje posee un hechizo equipado y pulsa el botón de cambio de hechizo lo equipa como secundario. De tener un hechizo secundario, este es equipado como primario.

**-Andar:** el personaje puede moverse horizontalmente con los botones de movimiento.

Para realizar estas acciones, por tanto, se necesitan tres botones de acción, uno de salto y dos de movimiento, una cantidad que entra dentro de lo común para este tipo de juegos y que no supone ninguna dificultad para el usuario medio perteneciente al grupo objetivo.

## 4. Especificación

A continuación se exponen los dos ejemplos más relevantes de la documentación desarrollada en las fases de análisis y diseño del proyecto: la tabla de requisitos y el diagrama de clases.

### 4.1. Tabla de requisitos

La figura 3 muestra la tabla de requisitos del juego. Contiene los principales requisitos que el proyecto deberá satisfacer así como su estado de completitud.

	Requisito	Estado
	<b>REGLAS DE JUEGO</b>	
RF1	El usuario deberá responder a un test a partir de cuyas respuestas se generará el argumento del juego y la secuencia de niveles que conformarán la partida.	Pendiente
RF2	El usuario deberá superar cuatro niveles consecutivos para terminar con éxito una partida.	Implementado
RF3	Cada nivel estará compuesto por una cuadrícula de entornos interconectados.	Implementado
RF4	Para pasar de un entorno a otro adyacente habrá que cumplir la condición necesaria: derrotar a todos los enemigos o accionar el interruptor que abre las puertas.	Implementado
RF5	El jugador deberá tener una barra de vida.	Implementado
RF6	Cuando la vida llegue a cero el personaje habrá perdido la partida y deberá volver a empezar.	Implementado
RF7	El jugador tendrá un nivel de cordura.	Pendiente
RF8	El jugador podrá perder o ganar cordura.	Pendiente
RF9	El nivel de cordura afectará a la jugabilidad.	Pendiente
RF10	Para superar el nivel actual el jugador deberá cumplir la misión del mismo y derrotar al jefe final si lo hubiese.	Parcial
RF11	La misión de un nivel podrá ser:  -Encontrar la salida.  -Encontrar un cierto número de objetos.  -Eliminar a cierto enemigo.	Parcial
	<b>JUGABILIDAD</b>	
RF12	El jugador podrá atacar.	Implementado
RF13	El jugador podrá saltar.	Implementado
RF14	El jugador podrá invocar hechizos.	Implementado
RF15	El jugador podrá llevar un arma equipada.	Implementado
RF16	Si el jugador intenta equipar una segunda arma soltará la ya equipada.	Implementado
RF17	El jugador podrá equipar un hechizo como primario.	Implementado
RF18	El jugador podrá equipar un hechizo como secundario.	Implementado
RF19	El jugador podrá alternar las posiciones de los hechizos equipados.	Implementado
RF20	Si el jugador intenta equipar un hechizo teniendo un hechizo primario equipado, soltará el hechizo	Implementado

	primario equipado.	
RF21	Los hechizos secundarios otorgarán una habilidad pasiva.	Implementado
RF22	Invocar los hechizos primarios supondrá un coste.	Pendiente
RF23	Los enemigos tendrán distintos comportamientos dependiendo de su tipología.	Parcial
RF24	El jugador tendrá a su disposición una amplia gama de armas y hechizos, más de veinte de cada tipo.	Parcial
	APLICACIÓN	
RF25	El jugador podrá modificar el volumen de la música y de los efectos de sonido por separado.	Implementado
RF26	El jugador podrá configurar los controles.	Parcial
RF27	El jugador podrá pausar el juego.	Implementado
RF28	El jugador podrá poner fin al juego en cualquier momento.	Implementado
RF29	El avance del jugador podrá desbloquear contenido para futuras partidas.	Parcial
	NO FUNCIONALES	
RNF1	La aplicación podrá ejecutarse en Windows.	Implementado
RNF2	La aplicación podrá ejecutarse en MacOS	Implementado
RNF3	Los entornos que componen los niveles deberán generarse procedualmente.	Implementado
RNF4	La generación de los entornos será de tal forma que permita siempre alcanzar las puertas y los objetos necesarios.	Implementado

*Figura 3. Tabla de requisitos.*

Los requisitos implementados definen la demo que se ha presentado a los usuarios en la prueba presencial explicada en el apartado 9.

## 4.2. Diagrama de clases UML simplificado

El diagrama de clases UML simplificado (que no contiene atributos ni métodos) de la figura 4 muestra las relaciones existentes entre las distintas clases implementadas durante el desarrollo del proyecto y su división en paquetes según su función y ámbito de trabajo.

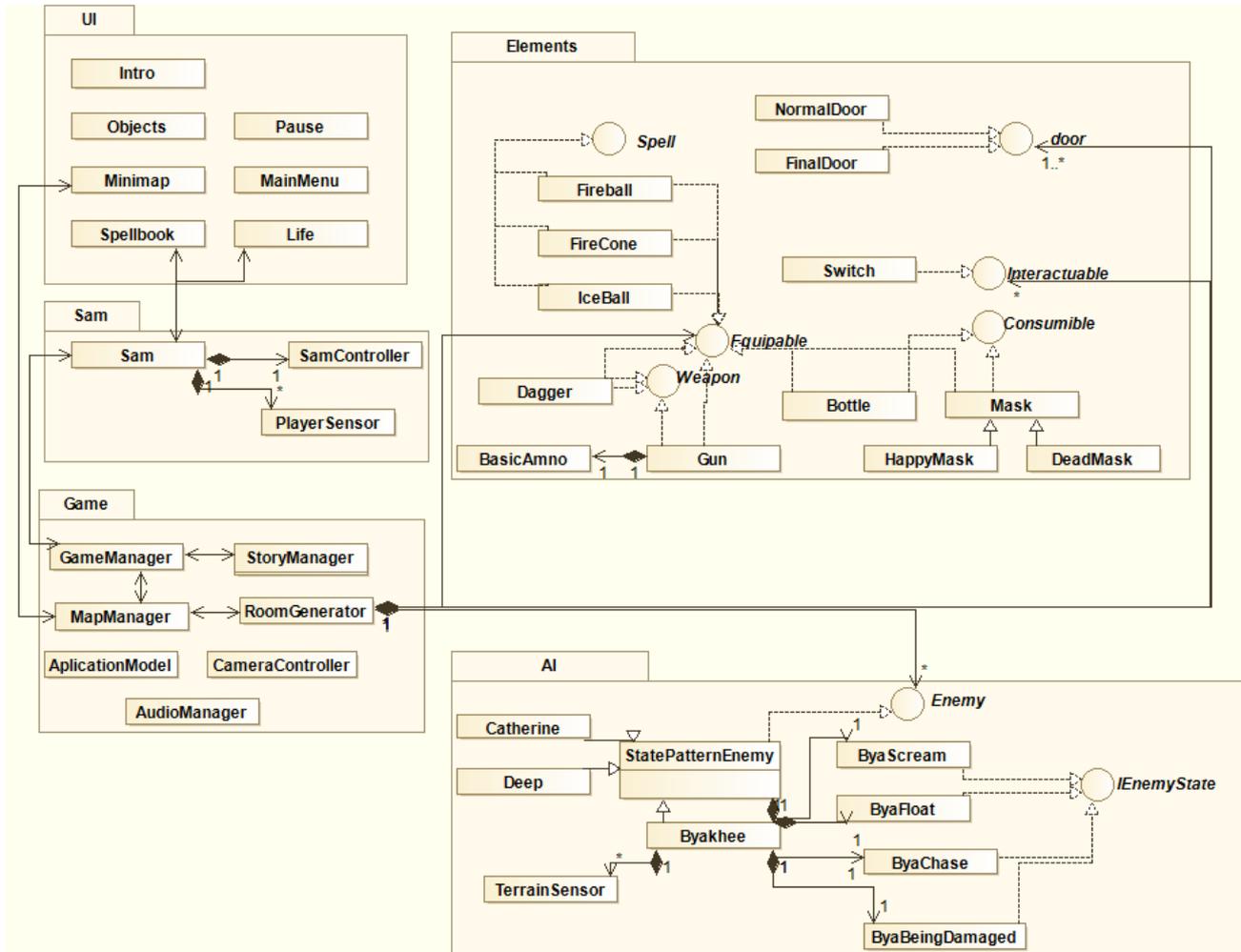


Figura 4. Diagrama de clases UML simplificado.

A continuación se explica brevemente el contenido de cada paquete y la función de cada clase implementada.

### UI

Contiene las clases que definen el comportamiento de los elementos de la interfaz gráfica y los distintos menús.

**Intro** - Implementa una cola de eventos que se sucederán al iniciar la aplicación tales como el título o el aviso de riesgo para la salud.

**Minimap** - Gestiona y actualiza el minimapa, ofreciendo información extraída del MapManager sobre la disposición de los entornos, su tipología y la posición actual.

**Pause** - Contiene las funciones necesarias para el funcionamiento del menú de pausa: controles de volumen y botones de opciones de audio, vuelta al menú principal y reanudación de la partida.

**MainMenu** - Programa el funcionamiento de los botones del menú principal: selección de ambiente, comienzo de partida y cierre de la aplicación.

**Life** - Gestiona y actualiza la barra de vida del personaje con información extraída de la clase Sam.

**Objects** - Muestra y actualiza el contador de objetos necesarios para finalizar la misión si la hay.

**SpellBook** - Gestiona el indicador del libro de hechizos; un elemento de interfaz con dos huecos para mostrar el icono de los hechizos equipados.

## Elements

Este paquete agrupa las clases relacionadas con los elementos que pueden aparecer a lo largo del juego.

**Spell** - Interfaz que define el comportamiento común a todos los hechizos. Un hechizo puede ser equipado como secundario, desequipado de esa posición, intercambiado y lanzado.

**Interactable** - Interfaz común a todos los objetos con los que el personaje puede interactuar. Un objeto interactuable puede ser consultado sobre su disposición a la interacción y puede ser interactuado.

**Consumible** - Interfaz para los objetos consumibles. Un objeto consumible puede ser consultado sobre su estado y puede producir un efecto al personaje.

**Equipable** - Interfaz que define el comportamiento común de los objetos equipables. Un objeto equipable puede ser equipado, desequipado y consultado sobre su estado.

**Weapon** - Interfaz común a todas las armas. Un arma puede atacar.

**Dagger** - Programa el comportamiento del arma básica cuerpo a cuerpo y sus atributos. Otras armas con características similares heredarán de ella.

**Gun** - Programa el comportamiento del arma básica de proyectiles y sus atributos; otras armas heredarán de ella.

**BasicAmno** - Contiene la información de la munición para un arma de proyectiles. Distintas municiones heredarán de ella.

**Mask** - Implementa un objeto de misión para la demo.

**Bottle** - Implementa el comportamiento de una poción curativa. Al consumirse cura al personaje.

**Switch** - Implementa el botón que sirve para abrir y cerrar las puertas de una sala.

**door** - Interfaz que define el comportamiento de una puerta. Una puerta puede ser consultada sobre su estado y atravesada.

**NormalDoor** - Implementa una puerta que al ser atravesada lleva a un entorno adyacente.

**FinalDoor** - Implementa una puerta que al ser atravesada lleva al siguiente nivel.

## Game

Contiene las clases relacionadas con la gestión y creación del contenido del juego y de la partida.

**GameManager** - Clase principal. Gestiona todos los elementos de una partida y coordina a las demás clases que intervienen en ella.

**StoryManager** - Se ocupa del test inicial y de la creación y gestión del arco argumental. Esta clase no se encuentra implementada en la demo y es la única diferencia respecto al juego completo.

**MapManager** - Contiene los métodos necesarios para la creación y gestión de un mapa o nivel.

**RoomGenerator** - Contiene los métodos para crear y devolver un entorno con las características solicitadas: tamaño, ambiente, algoritmo de generación, etc.

**ApplicationModel** - Gestiona las variables globales que permanecen entre escenas y ejecuciones.

**CameraController** - Se ocupa del control y configuración de la cámara del juego.

**AudioManager** - Gestiona la reproducción y opciones de audio del sistema.

## Sam

Reúne las clases que definen y controlan al personaje del jugador.

**Sam** - Almacena los atributos y equipo del personaje e informa sobre ellos a las clases que lo solicitan.

**SamController** - Gestiona el control del personaje y su comportamiento ante colisiones.

**PlayerSensor** - Informa a SamController de los objetos con los que el personaje entra en contacto.

## AI

Contiene las clases que se ocupan del comportamiento de los enemigos.

**Enemy** - Interfaz que define a un enemigo. Un enemigo puede recibir daño y cambiar de estado.

**IEnemyState** - Interfaz que define la funcionalidad de un estado de un enemigo.

**StatePatternEnemy** - Clase base de la que heredan todos los enemigos basados en máquinas de estados.

**Catherine** - Clase que define el comportamiento y atributos de los enemigos fantasmas.

**Deep** - Clase que define el comportamiento y atributos de los enemigos profundos.

**Byakhee**<sup>1</sup> - Clase que define el comportamiento y atributos de los enemigos byakhees.

**ByaScream** - Define el estado "gritando" de un byakhee.

**ByaFloat** - Define el estado "vagando" de un byakhee.

**ByaChase** - Define el estado "persiguiendo" de un byakhee.

**ByaBeingDamaged** - Define el estado "siendo herido" de un byakhee.

**TerrainSensor** - Sensor que informa a un enemigo cuando entra en contacto con elementos de terreno.

---

<sup>1</sup> Solo se describen los estados programados para el Byakhee para abreviar, pero Catherine y Deep tienen un número similar de clases estado.

## 5. Generación de entornos

Los niveles del juego están compuestos por entornos. Cada entorno es generado por un algoritmo que depende del resultado deseado. Estos algoritmos trabajan sobre una matriz de enteros que definirá la distribución de los distintos elementos en el propio entorno. En esta matriz un elemento a uno indica que en esa posición hay que colocar un bloque de terreno, uno cuyo valor sea cero indica un espacio vacío y se emplean otros enteros para puertas, enemigos, plataformas o mobiliario.

Una vez completada la matriz, con la forma y distribución del entorno, se recorre creando los elementos indicados en ella y almacenándolos en un vector de elementos que es devuelto al MapGenerator que lo almacena en su posición de la estructura mapa.

A continuación se da una breve descripción de cada algoritmo de generación, de su propósito y se muestran sus resultados.

### 5.1. Paso aleatorio en dos dimensiones modificado o paso borracho.

Este es el algoritmo más sencillo. El algoritmo de paso aleatorio<sup>[2]</sup>, *random walk*, se caracteriza porque en cada iteración se decide de forma aleatoria cual será el siguiente movimiento. En este caso cada dirección, ocho en total, tendrá la misma probabilidad de ser escogida a la hora de excavar el terreno.

Se ha modificado el algoritmo añadiendo una probabilidad de que en cada paso dado se comience un corredor, lo que equivale a dar un número aleatorio de pasos excavando en la misma dirección.

Además, como los enemigos y el jugador son más grandes que el bloque de terreno básico, en cada paso se excava una máscara de casillas en torno a la posición deseada en vez de una única casilla.

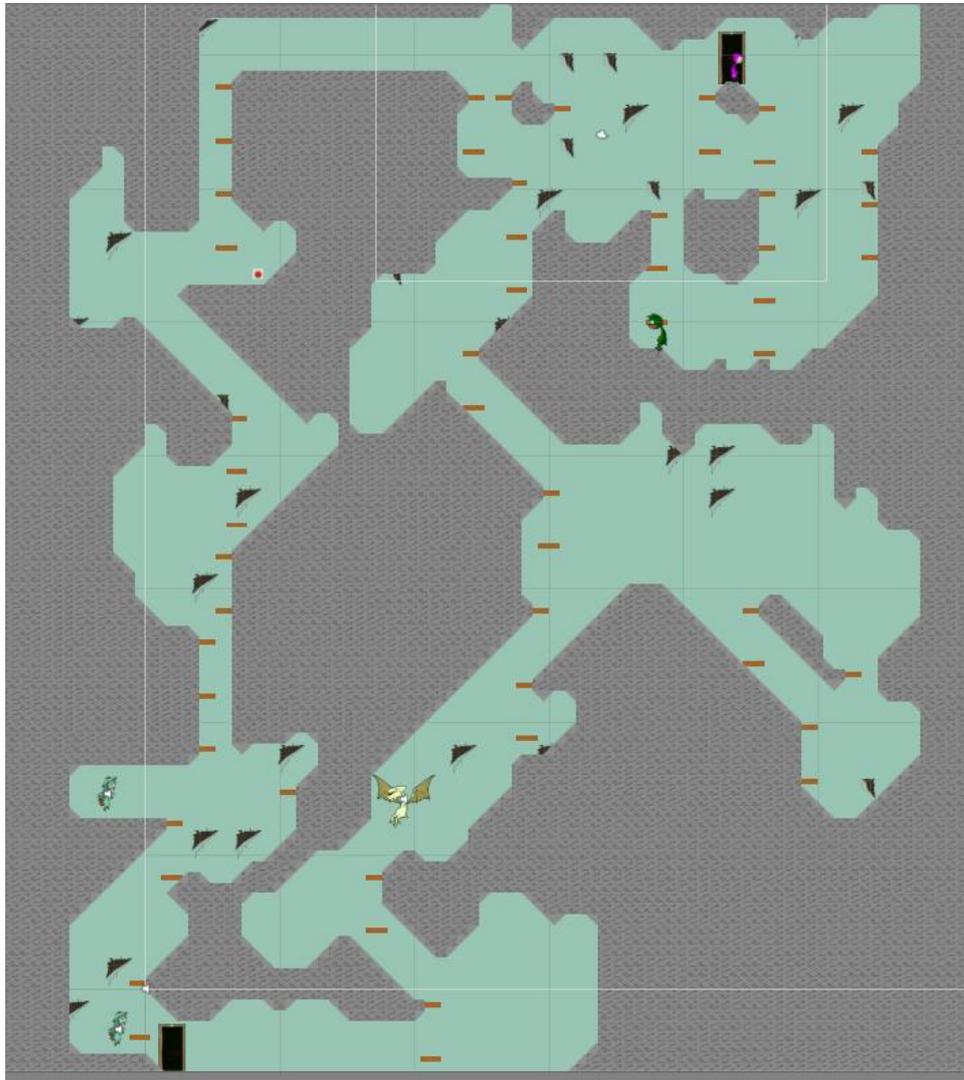
Inicialización.

- Se rellena toda la matriz con unos.
- Se indica que no se está en un corredor.

En cada iteración:

- Si no se está en un corredor se selecciona una dirección aleatoria como dirección actual.
- Si no se está en un corredor se decide aleatoriamente si empezar un nuevo corredor.
- Se da un paso en la dirección actual aplicando la máscara a la matriz, convirtiendo todos los unos tocados por la misma en ceros y contabilizándolos para la condición de fin.
- Repetir hasta alcanzar la condición de fin.

Con una condición de fin en torno al 30% de terreno excavado el algoritmo es rápido y genera pequeñas cavidades conectadas por corredores como puede verse en la figura 5. Este algoritmo se usa para la construcción de catacumbas y otros escenarios que requieran la combinación de grutas naturales con corredores artificiales.



*Figura 5. Mazmorra.*

## 5.2. Generación de laberintos

Existen diversos tipos de laberintos y distintos algoritmos para construir cada uno de ellos. Se ha decidido implementar el algoritmo de construcción de Prim para laberintos de conexión simple<sup>[3]</sup>, aquellos que no tienen bucles, por su eficacia y sencillez.

El algoritmo parte de una matriz de habitaciones y paredes y, una a una, añade las habitaciones al laberinto final quitando ciertas paredes. En la matriz las habitaciones se representan con valores a cero y las paredes con valores a uno.

Inicialización:

- Se selecciona aleatoriamente una habitación como inicial y se añade al conjunto I.

- Se añaden al conjunto F todas las habitaciones adyacentes a la habitación inicial.

En cada iteración:

- Se extrae aleatoriamente una habitación del conjunto F, se conecta con una habitación adyacente del conjunto I convirtiendo el valor uno de la pared que las separa en cero y se añade al conjunto I.

- Se añaden al conjunto F las habitaciones adyacentes a la habitación seleccionada que no estuviesen ya en F ni en I.

- Repetir hasta que el conjunto F quede vacío.

El algoritmo anterior genera laberintos con habitaciones de tamaño uno. Una vez obtenido dicho laberinto, se crea una matriz mayor en la que se traduce el laberinto generado a un laberinto con habitaciones del tamaño deseado. Los cálculos para obtener las dimensiones de la nueva matriz pueden consultarse en la bibliografía<sup>[3]</sup>.

Es en esta nueva matriz ampliada donde, según se van construyendo las habitaciones, se van añadiendo los demás valores que representan puertas, mobiliario, enemigos y demás elementos mediante probabilidades definidas por el ambiente.

Los entornos obtenidos con este algoritmo son perfectos para usarlos como interiores de edificios: templos, mansiones, etc. como se puede apreciar en la figura 6.



### 5.3. Autómatas celulares

Los autómatas celulares modelan de una forma sencilla comportamientos naturales de elementos que interactúan entre ellos a nivel local<sup>[4]</sup>. Este fenómeno es fácilmente trasladable al desarrollo de entornos naturales para hacer que el terreno "crezca".

Mediante una serie de reglas se hace que aparezcan o desaparezcan unidades de terreno en relación a los elementos que tienen a su alrededor. De esta forma paredes y túneles van creciendo y tomando forma en cada uno de los pasos hasta conformar los paisajes deseados<sup>[5]</sup>.

Este es el algoritmo más interesante, pero requiere que ciertos parámetros se obtengan por experimentación, como el número de iteraciones, y los umbrales de nacimiento y muerte de las células.

#### Inicialización:

- Se rellena la matriz del tamaño deseado con unos y ceros de forma aleatoria. La probabilidad de que un elemento sea uno dependerá del nivel de densidad deseado.

#### En cada iteración

- Se crea una nueva matriz.

- Para cada elemento de la matriz:

- Según las reglas definidas para el autómata, se calcula si debería mantenerse el valor actual o cambiarlo por otro. Es decir, si siendo cero debería nacer una nueva célula por estar rodeada del número suficiente de células o si siendo uno debería morir una célula por no tener suficientes células alrededor.

- Se copia el nuevo resultado a la nueva matriz.

- La nueva matriz sustituye a la matriz original.

Tras el autómata celular propiamente dicho se aplican a la matriz dos funciones, *fix* y *fix2*. *Fix* detecta las cavidades más pequeñas, aplicando un algoritmo de relleno por difusión<sup>[6]</sup>, y las elimina para dejar una única cavidad evitando que queden zonas sin comunicar. *Fix2* suaviza los bordes del perfil generado cambiando bloques cuadrados por medios bloques con pendiente y puebla el entorno con los demás elementos, entre ellos las plataformas que hacen accesibles todas las zonas generadas. Estas dos funciones se aplican también tras el algoritmo de paso borracho en la generación de mazmorras.

Este algoritmo es perfecto para generar grandes cavernas naturales con espacios y paredes irregulares como se puede ver en la figura 7. Además, con una pequeña modificación de las reglas del autómata, permite crear entornos naturales sin techo ni paredes que se pueden usar como exteriores como los de la figura 8. Para no tener isletas no conectadas con el suelo, se

aplica el algoritmo de relleno por difusión para localizar el suelo y se elimina todo el terreno que no haya sido rellenado.

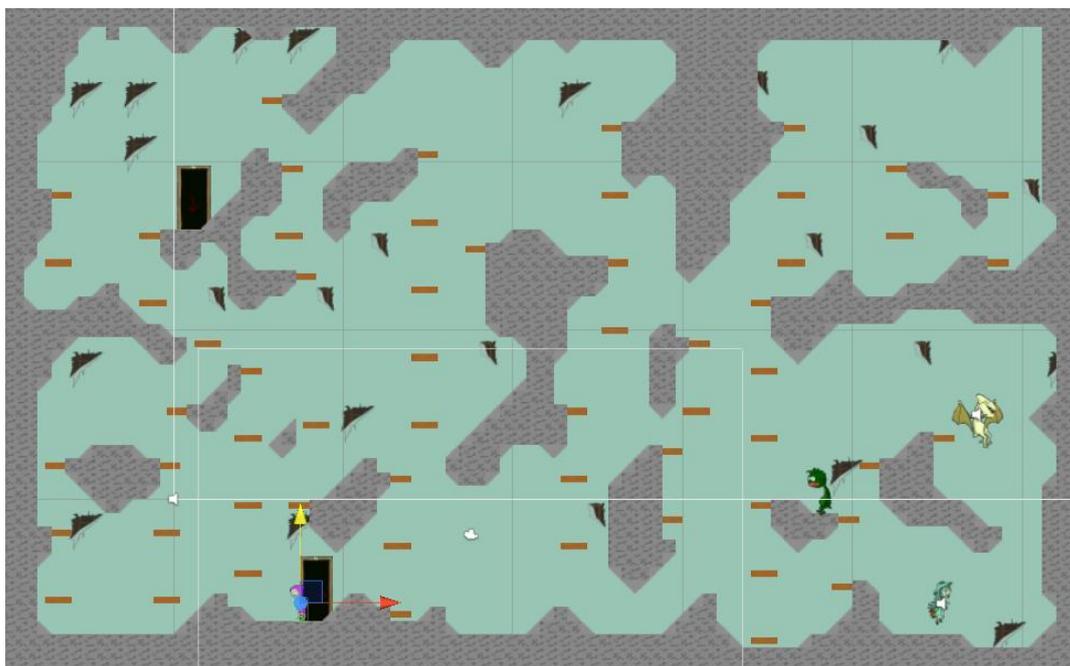


Figura 7. Caverna.

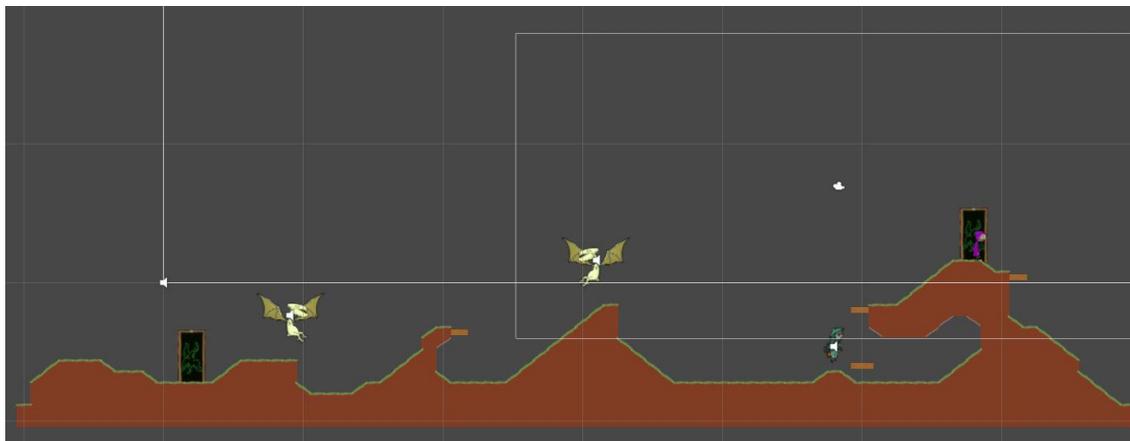


Figura 8. Valle.

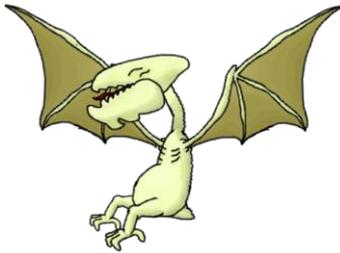
## 6. Enemigos

Una característica importante del proyecto es la variedad de enemigos y sus distintos comportamientos. Para la prueba con usuarios se han implementado tres enemigos que utilizan como base una máquina de estados. Cada enemigo tiene sus propios estados y diagrama de transiciones, como puede verse en el anexo D.



**Catherine:** la pequeña fantasma es el enemigo más sencillo, se dedica a vagar por el escenario, atravesando paredes y obstáculos hasta que ve al personaje. En ese momento se le iluminan los ojos en rojo y se dirige directa hacia él. Al ser herida avanza en línea recta durante un breve tiempo en el que no puede ser herida ni herir al jugador.

**Deeper:** el profundo es el enemigo más débil. Patrulla por el piso del escenario, tomándose sus descansos, hasta que ve al personaje y echa a correr hacia él. Si el personaje salta o sale de su campo de visión se desorienta y vuelve a la patrulla.



**Byakhee:** el enemigo más formidable y temido por los jugadores. El byakhee aletea por el escenario hasta que el jugador entra en su rango de visión. Cuando esto ocurre el byakhee suelta un grito de llamada que atrae a los demás de su especie hacia él y comienzan a perseguir al jugador. Para desplazarse por el escenario utiliza un movimiento heurístico que presenta diversas ventajas: logra que el byakhee no se atasque indefinidamente en un punto, evita una posible ralentización del sistema debido a la cantidad superior de cálculos que supondría la utilización de métodos más complejos, y además se consigue un movimiento menos mecánico. Esta heurística introduce la probabilidad de que algunos aleteos sean en una dirección aleatoria.

## 7. Herramientas

Para llevar a cabo el proyecto se ha optado por las siguientes herramientas:

### 7.1. Gimp

Gimp es una suite de dibujo libre y gratuita. Se ha elegido esta opción porque ofrece todas las funcionalidades requeridas por el proyecto para la edición de imagen sin encarecer el presupuesto final ni suponer problemas de licencias. Todos los componentes gráficos del juego han sido creados y editados con Gimp y son obra del autor.

### 7.2. Spriter Pro

Spriter Pro es un software de animación en 2D. Se ha elegido esta opción porque ofrecía la funcionalidad buscada de animación 2D mediante esqueletos rígidos a un precio de licencia muy razonable en comparación con programas similares.

Cada elemento animado ha sido compuesto por una veintena de piezas independientes, dibujadas en Gimp, articulado por un esqueleto rígido, animado y renderizado en una serie de sprites estáticos que serán usados por Unity.

### 7.3. Unity

Unity es un motor y entorno de desarrollo de videojuegos que ofrece licencia gratuita si se cumplen ciertos requisitos. Uno de los mayores puntos a favor de esta opción es la cantidad de documentación y tutoriales disponibles así como la gran comunidad de desarrolladores que hay en torno a él.

El uso de este motor elimina la necesidad de programar ciertos aspectos a bajo nivel permitiendo al desarrollador centrarse en el diseño de los elementos del juego y su comportamiento mediante scripts. Aun así, también permite profundizar todo lo que se desee para conseguir ciertos comportamientos o funcionalidades específicas con el fin de conseguir resultados mucho más personalizados.

## 8. Resultados

A continuación se muestran capturas de la demo para ilustrar las principales características implementadas.

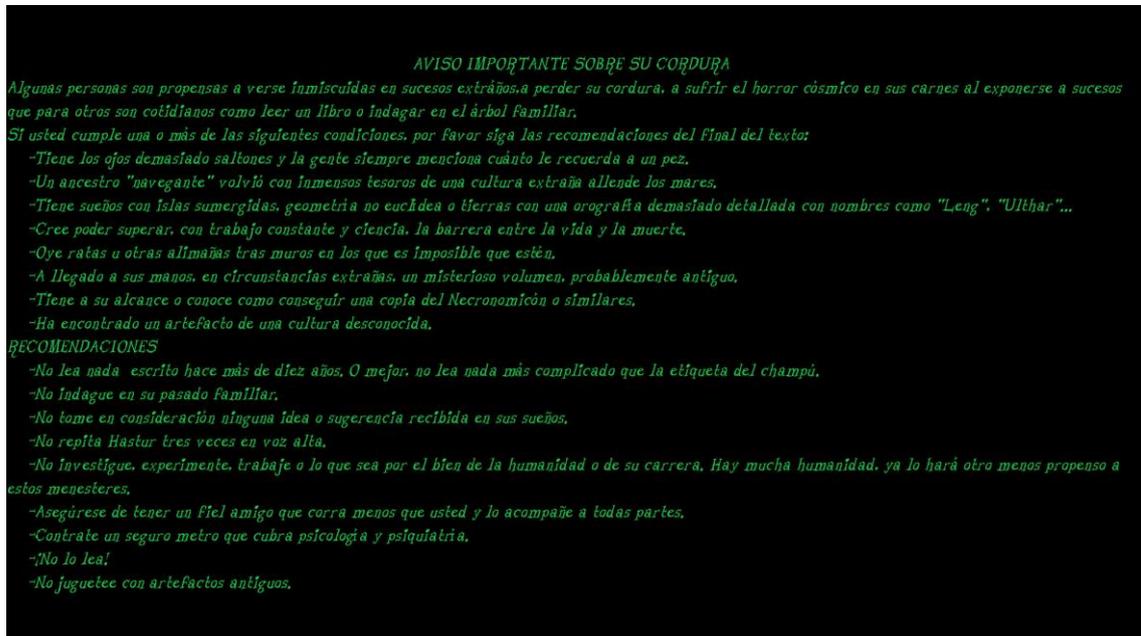


Figura 9. Aviso de salud mental.

La figura 9 muestra el aviso sobre salud mental, un guiño en clave de humor a las múltiples causas de locura en la obra de Lovecraft.



Figura 10. Menú principal.

La figura 10 muestra el menú principal de la demo con el botón de inicio de partida, los botones de selección de nivel y el botón para salir de la aplicación.

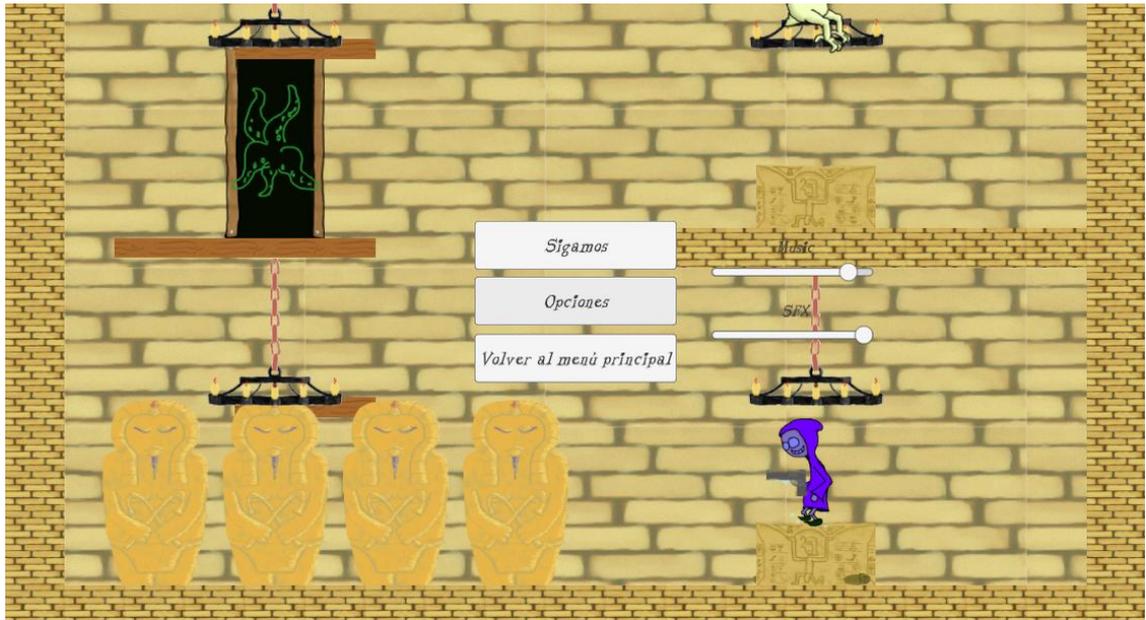


Figura 11. Pausa.

La figura 11 muestra el juego pausado y los botones del menú de pausa. El primer botón cancela la pausa y reanuda la partida, el segundo muestra u oculta las barras de configuración de las opciones de sonido y el tercero termina la partida y vuelve al menú principal del juego.

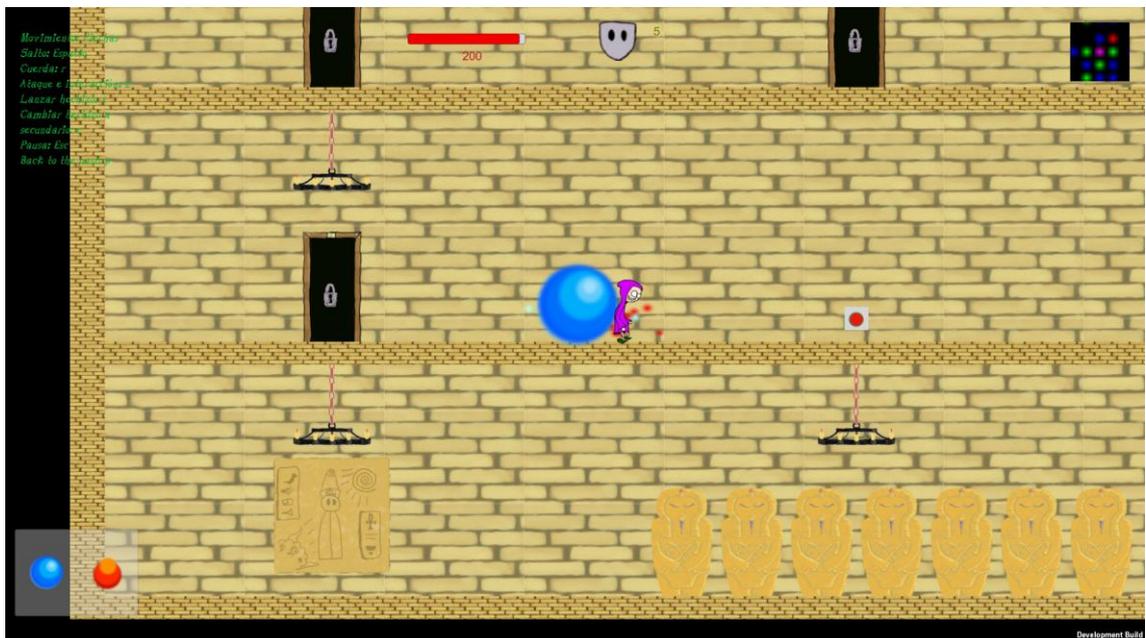


Figura 12. A pulsar el botón.

En la figura 12 se aprecia el interior de una sala de la que solo se puede salir pulsando el botón rojo. Abajo a la izquierda se puede ver el libro de hechizos, el elemento de la interfaz gráfica que informa de los hechizos equipados, en este caso la bola de hielo como primario y la bola de fuego como secundario. Se puede ver también la barra de vida y el contador de objetos de misión con forma de máscara.

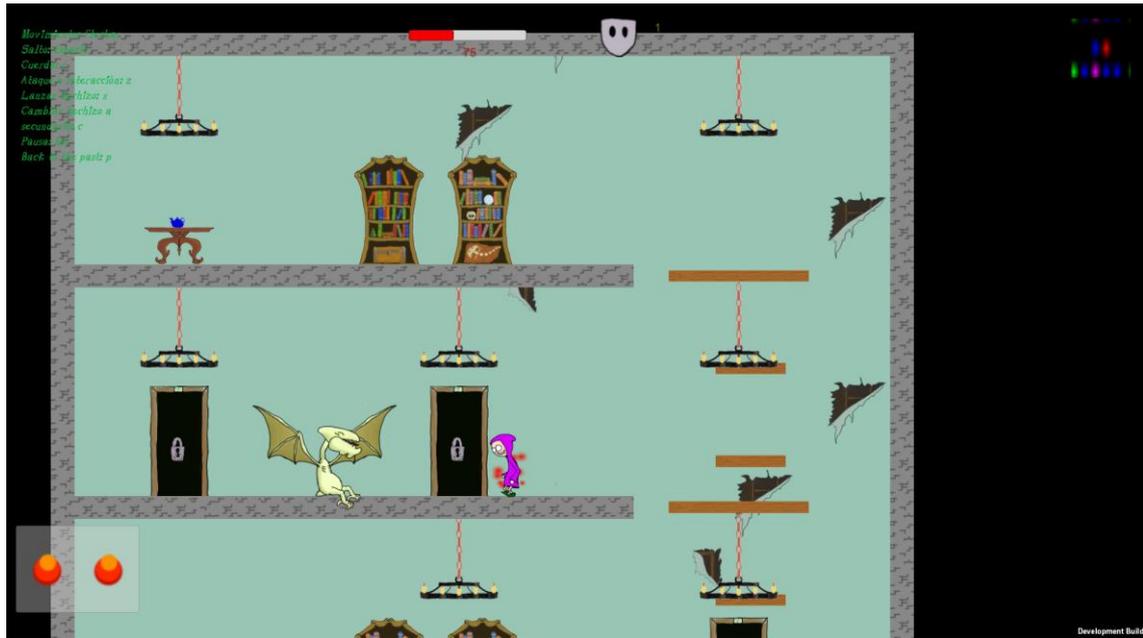


Figura 13. Contra el byakhee.

En la figura 13 se puede ver el interior de la mansión y a Sam, el sectario, plantando cara a un Byakhee que le ha quitado más de media vida. Arriba a la derecha se ve la distribución de entornos en el minimapa.

## 9. Validación y comunicación con usuarios

Una parte importante de este proyecto es la interacción y opinión de los usuarios. Para establecer este contacto y obtener información se han tomado tres medidas: el diseño y distribución de un formulario online sobre contenido, la creación de un grupo de pruebas y la celebración de dos sesiones de pruebas presenciales con jugadores pertenecientes al grupo objetivo.

### 9.1. Formulario

Se ha diseñado y publicado en redes sociales un formulario, ver anexo B, en el que los usuarios pueden dar a conocer lo que esperan de un producto de las características de "Descenso". Qué aspectos jugables prefieren, qué partes de la ambientación piensan que son más importantes, qué elementos esperan encontrar, etc. Hasta el momento de la escritura de este trabajo han respondido 28 personas.

De este formulario se han obtenido resultados tan interesantes como que los usuarios prefieren jugar en la segunda década del siglo XX, que quieren poder visitar una mansión encantada, las tierras del sueño y una pequeña localidad decadente, o que les gustaría poder controlar a un catedrático de geometría no euclídea. En el anexo B se muestran todos los resultados.

### 9.2. Grupo de prueba

Se ha escogido a un pequeño grupo de personas, dos graduados en informática y tres estudiantes de último año de distintas ingenierías, y se les ha dado acceso a una carpeta compartida. En esta carpeta se publica un ejecutable cada vez que se añade algún nuevo contenido o funcionalidad. De esta forma se pretende mantener un desarrollo ágil dirigido por pruebas<sup>[7]</sup>. También se les ha dado acceso a un documento compartido en el que publicar los errores que encuentren o las sugerencias que les puedan surgir. Gracias a este método se han detectado, entre otros, los siguientes casos:

- Se generaba terreno demasiado cerca del borde superior impidiendo acceder a ciertas zonas.
- El personaje no podía girar en salto y el grupo pensó que era deseable que sí pudiera.
- Algunos sonidos no se veían modificados al modificar las opciones de audio.
- El personaje atravesaba el techo al trepar por cuerdas.
- Algunos enemigos no recibían daño.
- Algunos hechizos se perdían tras un uso.

### 9.3. Pruebas presenciales

Como se ha comentado anteriormente se han realizado dos sesiones de pruebas presenciales con un grupo de usuarios. Este grupo estaba compuesto por 14 personas de 18 a 28 años y con experiencia media o alta en videojuegos.

La primera prueba fue una toma de contacto con los usuarios y, aunque estuvo mucho menos planificada que la segunda, sirvió para obtener información muy importante. Se trataba de poner a prueba el sistema de generación de mapas y entornos así como el control del personaje principal. De entre las conclusiones obtenidas en esa prueba cabe destacar:

- La velocidad de caída del personaje resultaba incómoda a los jugadores. Así que se aumentó la gravedad sobre el mismo.

-Hacía falta que el salto fuese adaptativo.

-Algunas plataformas no se generaban correctamente, quedando lugares inaccesibles, sobre todo cerca de los bordes.

Durante la segunda prueba se propuso un objetivo claro a los jugadores, superar un nivel del juego. Para conseguirlo tenían que encontrar un número determinado de máscaras y alcanzar la puerta de salida. Además, se les hizo rellenar un formulario sobre jugabilidad y satisfacción.

A pesar de que más del 75% necesitó varios intentos, todos mostraron motivación por conseguir el objetivo, como reflejaron en la encuesta, y todos lo lograron tarde o temprano sin mostrar signos de aburrimiento. A continuación se listan algunas de las conclusiones más importantes tras la prueba y en el anexo C se muestran todos los resultados del formulario.

-Había que aumentar las cajas de colisiones de aquellos objetos con los que se puede interactuar.

-Era necesario programar un tiempo de invulnerabilidad al entrar en una sala para no ser atacado nada más aparecer sin poder reaccionar.

-Algunos hechizos no funcionan como se esperaba.

-En un caso muy concreto la puerta de salida no aparecía.

Gracias a estas dos pruebas se encontraron errores de diseño o programación que hasta el momento el grupo de pruebas había pasado por alto, debido a su familiaridad con el proyecto.

## 10. Conclusiones

Como se menciona al principio del documento, el objetivo de este proyecto es desarrollar y comercializar un videojuego independiente. Dada la envergadura del mismo, para este trabajo fin de grado se ha decidido limitar el objetivo a la obtención y prueba con usuarios de una demo funcional que refleje las características del producto final. Así pues, se ha implementado la demo de "Descenso", un plataformas procedural ambientado en los mitos de Lovecraft, y ha sido puesta a prueba obteniendo buenos resultados y un importante *feedback* de los usuarios.

A nivel técnico, la demo, desarrollada en Unity, implementa tres algoritmos de generación de entornos:

- Paso borracho: un algoritmo de *random walk* que, pese a su sencillez, presenta unos resultados muy llamativos y útiles para generar complejos de cuevas interconectadas mediante pasadizos.

- Generación de laberintos de conexión simple de Prim: los laberintos obtenidos por este algoritmo, una vez ampliados, son perfectos para la representación del interior de edificios. Esta ampliación, y la casuística que genera el tamaño variable de las habitaciones, fue el primer escollo del proyecto; estos laberintos se suelen diseñar para su recorrido en el plano horizontal y en esta caso tenía que poder ser recorrido en el plano vertical por un personaje con una capacidad de salto limitada.

- Autómata celular: la capacidad de simular sistemas naturales de este tipo de algoritmos ha demostrado ser perfectamente aplicable a la generación de entornos resultando en cavernas o exteriores con irregulares formaciones de terreno. De los tres algoritmos de generación de entornos fue el que más tiempo costó implementar, no tanto por la dificultad de su implementación, sino por la necesidad de calibrar, mediante experimentación, ciertos parámetros como los umbrales de generación o el número de iteraciones. Además, junto a este algoritmo se desarrollaron las funciones, aplicadas después al paso borracho, que suavizan el perfil del escenario y aseguran su correcta conexión y recorrido mediante la eliminación de pasadizos no conectados al espacio principal y la adición de plataformas.

También se han implementado tres tipos de enemigos modelados mediante máquinas de estados. El mayor desafío en su desarrollo fue conseguir un movimiento lo más natural posible sin invertir demasiado tiempo de cálculo. Esto se consiguió implementando movimientos heurísticos que, además, impiden que los enemigos queden atascados indefinidamente en elementos del terreno.

Durante el periodo de desarrollo se ha mantenido un contacto directo con los usuarios en todo momento y se han realizado, además, dos sesiones de prueba presenciales. Esta comunicación y pruebas han revelado los dos mayores retos encontrados durante este proyecto: los controles del personaje y la disparidad de opinión entre usuarios. Ambos problemas están relacionados, pues uno de los aspectos en los que los usuarios mostraban desacuerdo era en la forma en que el personaje debía responder a sus órdenes. Ante estas disyuntivas se ha intentado identificar a los usuarios con mayor experiencia o formación en el campo para dar más peso a sus sugerencias.

De las conclusiones extraídas de todas las pruebas con usuarios, se muestra que se han alcanzado los objetivos de motivación y dificultad buscados para el juego; pero también que queda mucho camino por recorrer si se quiere que este llegue al mercado.

Por último, este trabajo ha servido para comprobar personalmente la titánica tarea que supone el desarrollo de un videojuego, por modesto que este sea, lo gratificante que resulta obtener resultados y el valor que la comunicación con la comunidad puede ofrecer durante muchas fases del desarrollo. Supone el punto y final a la vida académica y, a su vez, abre la puerta a un nuevo capítulo, a afrontar el futuro con ilusión, luchando por llevar el proyecto hasta su objetivo final: la publicación de la versión completa de "Descenso".

## 11. Trabajo futuro

Quedan dos hitos importantes para completar "Descenso". El primero es conseguir financiación mediante la publicación y compartición de la demo fruto de este trabajo fin de grado y el segundo completar el videojuego. A continuación se describe cómo se piensan llevar a cabo.

### 11.1. Campaña de financiación colectiva y distribución digital

Se pretende buscar financiación mediante una campaña en Kickstarter (<https://www.kickstarter.com/>), la mayor plataforma de financiación colectiva de internet, y ofrecer el juego como recompensa a los mecenas que hagan una aportación mínima similar al precio de mercado estimado del producto final. Para que la campaña resulte atractiva la demo se distribuirá de forma gratuita.

El siguiente paso será intentar publicar el juego en Steam mediante su plataforma GreenLight. GreenLight ofrece la posibilidad de que desarrolladores independientes compartan sus proyectos y que estos sean valorados por la comunidad; si se obtienen los suficientes votos el proyecto pasa a formar parte del catálogo de venta de la tienda.

### 11.2. Completar el videojuego

Para que el juego esté completo, una vez conseguida la financiación, hay que añadir los siguientes elementos y funcionalidades:

#### **-Test inicial y generador de argumentos**

La demostración jugable ofrece dos únicos niveles a elegir por el usuario con una única misión: encontrar las X máscaras de la locura. La versión final debe realizar un test al jugador a partir del cual generar un arco argumental que le guíe por distintos niveles con distintas misiones.

#### **-Sistema de cordura**

El sistema de cordura explicado en la sección 3.5 aún está por implementar.

#### **-Ambientes**

La demo contiene la mansión y el templo; el producto final debe contener al menos seis más para lo que se tendrán en cuenta los resultados obtenidos por el formulario mencionado en la sección 8.1.

Hay que diseñar y dibujar los elementos que compongan estos entornos de forma que puedan ser usados por los algoritmos de generación ya implementados.

#### **-Enemigos**

Se han implementado tres enemigos que aparecen en los dos ambientes indistintamente. La versión final debe contener, al menos, dos enemigos propios de cada ambiente además de un conjunto de al menos cinco enemigos comunes.

#### **-Jefes**

Hay que diseñar e implementar los jefes que deben aparecer al final de cada nivel. Harán uso de la misma arquitectura que los enemigos comunes, la cual ya está implementada.

#### **-Hechizos y armas**

La demo contiene dos armas, daga y pistola, y tres hechizos, bola de fuego, cono de fuego y bola de hielo. El juego final dispondrá al menos de 20 hechizos distintos, de entre los cuales otros seis ya están diseñados, y de al menos otras 20 armas distintas.

### **-Consumibles**

Además de la poción de vida, que en la prueba aparece al eliminar a los Byakhees, la versión final contendrá otros consumibles que afectarán a otros atributos del personaje como puedan ser su velocidad, fuerza o nivel de cordura, o le otorgarán diversos estados.

## 12. Diagrama temporal

El diagrama de Gant de la figura 14 muestra la división en tareas del proyecto a lo largo del tiempo.

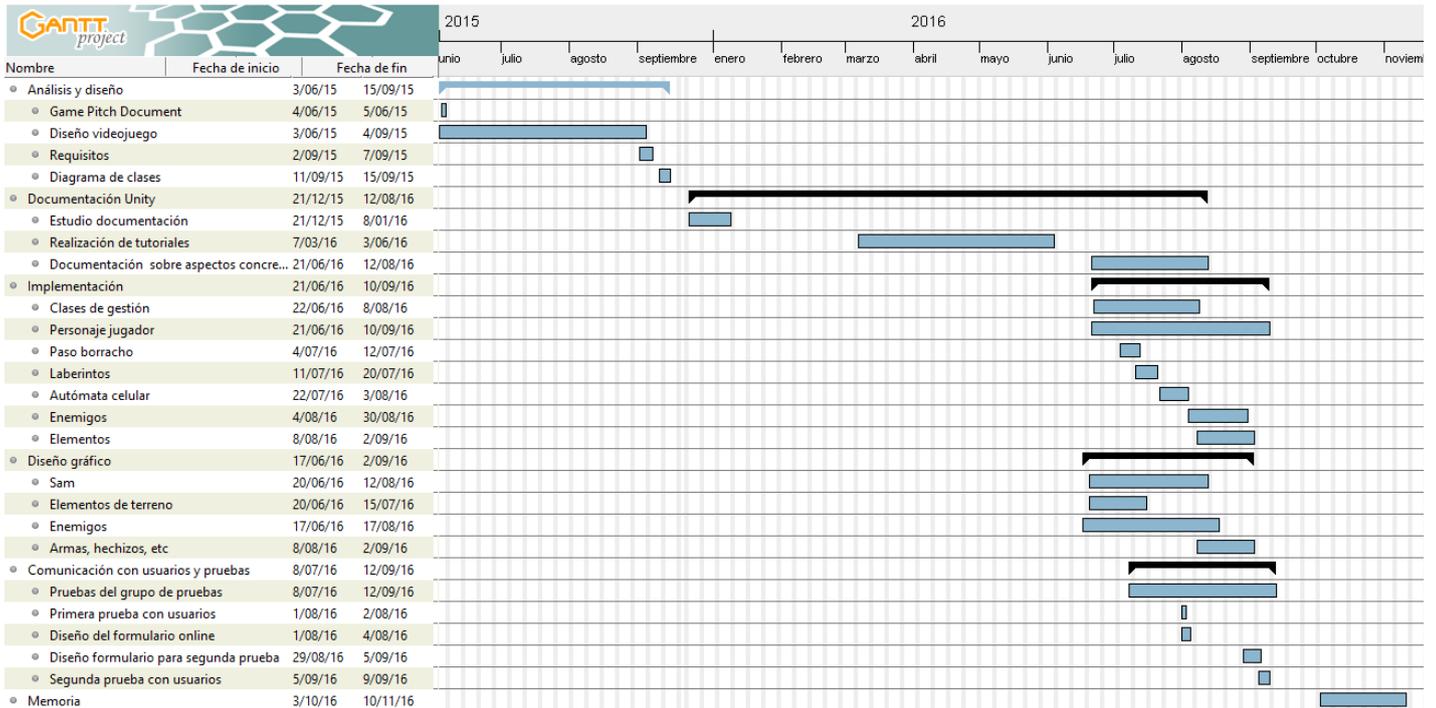


Figura 14. Diagrama de Gant.

El primer paso del proyecto fue diseñar el *game pitch document* que resumía a grandes rasgos las características del videojuego propuesto y que puede ser consultado en el anexo A. Una vez aprobado ese documento, se profundizó en el diseño del videojuego, sus reglas y características, haciendo diversos cambios a la idea inicial. Tras decidir todos los aspectos del juego se tradujeron a la tabla de requisitos y al diagrama de clases descritos en la sección 4 de este documento.

Después de especificar el proyecto, se abrieron tres ramas paralelas que en un estudio con un equipo mayor hubieran pertenecido a profesionales distintos. Así pues, se empezó con el diseño gráfico, la implementación y la comunicación con los usuarios.

La implementación se realizó de forma secuencial, añadiendo funcionalidades al proyecto. Se hizo un pequeño esqueleto de lo que sería el jugador y las clases de gestión y se fueron añadiendo los distintos algoritmos de generación, los enemigos y los elementos del juego.

En cuanto al diseño gráfico, se partió de unos diseños iniciales y después se fue produciendo el material según se iba necesitando en la rama de implementación.

Además, mientras tanto, había que mantener comunicación con el grupo de prueba para ir solucionando los problemas que iban surgiendo y no arrastrarlos hasta cuando fuese más difícil localizarlos.

Con el proyecto casi finalizado se realizó la segunda sesión de pruebas y con los resultados de esta se hicieron las últimas correcciones.

Por último, se ha redactado esta memoria y se ha elaborado el material gráfico que en ella aparece, tablas y capturas.

En la figura 15 se ve el desglose en horas de las distintas tareas.

<b>Análisis y diseño</b>	Total	38	Estimado	35
GPD		5		
Diseño videojuego		17		
Requisitos		5		
Diagrama de clases		11		
<b>Documentación de Unity</b>	Total	22	Estimado	20
Estudio documentación		10		
Tutoriales		12		
<b>Implementación</b>	Total	212	Estimado	170
Clases de gestión		18		
Personaje jugador		20		
Paso borracho		11		
Laberintos		34		
Autómata celular		49		
Enemigos		68		
Elementos		12		
<b>Diseño gráfico</b>	Total	44	Estimado	
Sam		5		
elementos de terreno		6		
Enemigos		15		
Armas, hechizos, etc		2		
Animación		16		
<b>Comunicación con usuarios y pruebas</b>	Total	37	Estimado	50
Pruebas		17		
Primera sesión de pruebas		4		
Diseño del formulario online		7		
Diseño del formulario para segunda prueba		3		
Segunda sesión de pruebas		6		
<b>Memoria</b>	Total	33	Estimado	25

Total	386	Total estimado	300
Sin gráficos	342		

Figura 15. Desglose en horas

### 13. Bibliografía

[1] H.P.Lovecraft, traducción de Francisco Torres Oliver, José María Nebreda y Juan Antonio Molina Foix

*Narrativa completa /vol 1 H.P.LOVECRAFT*

Valdemar 4ª ed: Madrid 2008

[2] Colaboradores de Wikipedia.

*Camino aleatorio* [en línea]. Wikipedia, La enciclopedia libre, 2016

[fecha de consulta: 8 de noviembre del 2016].

Disponible en <[https://es.wikipedia.org/w/index.php?title=Camino\\_aleatorio&oldid=94626218](https://es.wikipedia.org/w/index.php?title=Camino_aleatorio&oldid=94626218)>

[3] Víctor Tomás Tomás Mariano, Mariano Pozas Cárdenas, Jorge Hernández Camacho

*Propuesta para la generación de laberintos ampliados en 2D*

Artículo de la universidad autónoma del Estado de Hidalgo.

[fecha de consulta: 8 de noviembre del 2016].

Disponible en <http://www.uaeh.edu.mx/scige/boletin/huejutla/n1/a4.html>

[4] Colaboradores de Wikipedia.

*Autómata celular* [en línea]. Wikipedia, La enciclopedia libre, 2016

[fecha de consulta: 8 de noviembre del 2016].

Disponible en <[https://es.wikipedia.org/w/index.php?title=Aut%C3%B3mata\\_celular&oldid=90153452](https://es.wikipedia.org/w/index.php?title=Aut%C3%B3mata_celular&oldid=90153452)>.

[5] Michael Cook 2013

*Generate Random Cave Levels Using Cellular Automata*

[fecha de consulta: 8 de noviembre del 2016].

[6] Colaboradores de Wikipedia.

*Algoritmo de relleno por difusión* [en línea]. Wikipedia, La enciclopedia libre, 2015

[fecha de consulta: 8 de noviembre del 2016].

Disponible en <[https://es.wikipedia.org/w/index.php?title=Algoritmo\\_de\\_relleno\\_por\\_difusi%C3%B3n&oldid=87006127](https://es.wikipedia.org/w/index.php?title=Algoritmo_de_relleno_por_difusi%C3%B3n&oldid=87006127)>.

[7] D. Vallejo et al.

*Desarrollo de videojuegos. Un enfoque práctico. 24.3 TDD*

EdLibrix Julio 2014

## Anexo A. GPD

Morbi efficitur nisi a velit tincidunt pharetra. Pellentesque et augue venenatis, facilisis est nec, aliquet ipsum. *Ph'inglai Mglw'naft Cthulhu Rlyeh wgal'nagl fhtagn.* Donec consequat orci id aliquam varius. In commodo in justo sed fringilla. Curabitur at est placerat, tempus sem efficitur, imperdiet nisi. Integer eros urna, suscipit non rutrum sit amet ultricies vel nisl. Curabitur pharetra at nisi et laoreet. Quisque cursus dignissim lectus at magna dignissim, ut rutrum quis scelerisque fringilla, turpis ante vestibulum. In feugiat sit amet erat ut pharetra hendum. Nullam sit amet dapibus nisi. Phasellus justo felis placerat quam, ac efficitur est neque ac suscipit. Morbi commodo venenatis nisl, at scelerisque nisl pretium ac. Sed vel tellus a felis elementum consectetur ut quis arcu. Suspendisse potenti. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nulla facilisi. Aliquam lobortis purus sit amet turpis scelerisque dictum. Phasellus sapien velit, rutrum a enim eu, viverra semper velit. Aenean feugiat eros a massa dapibus, eu suscipit urna rutrum. Cras ornare neque sem, convallis porttitor lacus viverra et. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Ut at ante lectus. Nunc vitae viverra libero. *Ja Ja shub niggurath!* Curabitur at fermentum nunc. Nulla id tincidunt nisi. Proin justo ex, vestibulum et odio vel, rutrum maximus leo. Aliquam tempus lacus ut ipsum euismod, at tincidunt ipsum tristique. *Ph'inglai Mglw'naft Cthulhu Rlyeh wgal'nagl fhtagn.* Duis tincidunt sem sed dui porta, eu posuere mauris porttitor. Nulla felis erat, dignissim in commodo id, posuere in quam. Proin fringilla, orci eu pellentesque laoreet, nulla eros luctus nisi, eget consectetur arcu sapien vehicula mi. Etiam suscipit placerat lacinia. Aenean dapibus nunc id

### Descenso

### Game Pitch Document

David Orera Hernández

*“Que no está muerto lo que yace eternamente, y con los eones extraños incluso la muerte puede morir»*

Meses rebuscando en viejos archivos, persiguiendo a ariscos eruditos avaros de conocimientos prohibidos. He perdido mi trabajo, mis amigos, y dicen que mi cordura en la búsqueda insana de aquello que me la pueda devolver.

Mi formación académica me impedía creer en todo lo que ahora persigo, si no fuese porque he visto... porque sé...

Estaba a punto de rendirme cuando he encontrado lo que parece una lista inocua y un extraño diagrama. Pero estos elementos, en las proporciones adecuadas, dispuestos según dicta el diagrama, me pondrán en contacto, por fin, con el ente capaz de devolverme aquello que mi alma anhela. Cualquier precio, cualquier sacrificio, me parecerá justo si puedo contemplarla una vez más.

Descenso es la historia de alguien dispuesto a sacrificarlo todo, hasta su propia humanidad, para conseguir su objetivo.

Este alguien será un sectario, una persona normal que tras descubrir que existen poderes superiores ha decidido dedicar su vida a servirles para obtener algo a cambio, sea cual sea el coste.

Descenso pretende aportar el componente argumental que falta a los juegos procedurales estilo “rogue-like”.

## **Mecánica:**

El jugador elegirá la profesión que tenía su personaje antes de convertirse en un sectario.

Como no es lo mismo un bibliotecario que por azar encuentra un volumen arcano que un investigador privado que una noche se enfrenta a una criatura sobrenatural y después decide investigar esta elección determinará los atributos y equipo de nuestro sectario.

Después el jugador responderá a un pequeño cuestionario cuya pregunta más importante será: ¿Qué deseas? Esta y el resto de las preguntas servirán para “fijar” el itinerario del juego (Más adelante se explicará la necesidad de las comillas).

Una vez tengamos nuestro personaje y su motivación comenzará el juego que consistirá en ir avanzando por distintas mazmorras generadas proceduralmente en busca de hechizos y artefactos que nos ayuden a contentar a nuestro dios y en última instancia lograr nuestra meta. Estas mazmorras tendrán un marcado componente de plataformas y además estarán plagadas por multitud de criaturas procedentes de los mitos de Cthulhu. Para enfrentarse a los desafíos el protagonista llevará siempre un arma y hasta dos hechizos distintos así como diferentes artefactos que le proporcionarán bonificaciones.

Cada cierto número de fases el jugador deberá enfrentarse a un enemigo jefe de gran tamaño.

## **Armas:**

El jugador irá encontrando distintos tipos de arma a lo largo del juego, desde simples dagas hasta escopetas o cetros mágicos. Solo podrá llevar una a la vez.

## **Hechizos:**

El jugador podrá equipar hasta dos hechizos, uno de cada tipo. Los hechizos podrán ser ofensivos o de manipulación. Permitiendo estos últimos cambiar el entorno o la jugabilidad.

## **Artefactos:**

Este tipo de objetos otorgarán bonificaciones pasivas al personaje y podrá portar todos los que encuentre.

## **Sacrificio:**

Se ha hecho bastante hincapié en el coste o sacrificios que el personaje ha de hacer para lograr su objetivo. El que más impacto tendrá será el sacrificio de su cordura. Según se adentre en los mitos de Cthulhu y su comprensión el personaje se irá volviendo loco. La locura incrementará sus poderes mágicos, pero también le irá creando inconvenientes en forma de enfermedades mentales.

Además es posible que se le propongan situaciones en las que deba renunciar a parte de su salud física, equipo u otras alternativas.

## **Decisiones:**

De vez en cuando el jugador tendrá que tomar alguna decisión que influirá en su itinerario final, además este itinerario tendrá siempre un cierto componente aleatorio, así el jugador nunca puede estar seguro de que unas elecciones le vayan a llevar a un desenlace o camino ya conocido.

## **¿Por qué los mitos?**

Los mitos de Cthulhu ofrecen una gran ambientación que cuenta con un número de seguidores abismal. Sólo hay que echar un ojo a Kickstarter y se ve que cualquier proyecto que contenga Cthulhu en su nombre se financia en muy poco tiempo. Además ofrece un bestiario casi inagotable para poder dotar al juego de una gran variedad de antagonistas con formas y comportamientos muy bien definidos y que el jugador podrá reconocer fácilmente.

## Aspecto

Descenso será un juego en 2D con personajes animados mediante rotaciones en articulaciones. Para crear contraste con lo oscuro de la temática el juego será representado en su totalidad con recortes de cartulina de colores.

## Escenario:

El personaje viajará por escenarios muy distintos gráficamente:

- Cuevas.
- Regiones árticas.
- Las tierras del sueño.
- Otras realidades.

## Itinerarios

Todos los puntos del itinerario (escenarios, enemigos finales y primigenio) tendrán un factor aleatorio que variará según las respuestas al cuestionario inicial y a las decisiones del jugador a lo largo del juego.

## Primigenio

El "dios" al que el sectario sirve. Vendrá condicionado por las elecciones del cuestionario inicial e influirá en el resto del itinerario. El fin del juego será casi siempre un enfrentamiento con dicho primigenio y la condición de victoria será derrotarle (en casi todos los casos).

## Enemigos finales/jefes

Enemigos peligrosos y en general de gran envergadura cuyo enfrentamiento con éxito aportará al jugador una recompensa. Estos enemigos dependerán del primigenio y aparecerán al final de ciertas fases.

## ¿Por qué TFG?

El trabajo de fin de grado puede que sea la última oportunidad que tenga para hacer algo que de verdad me apasione. Una vez acabe la carrera quizá acabe en una empresa y allí habrá poco de creatividad...

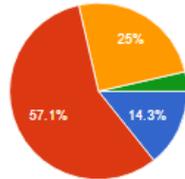
Siempre me ha apasionado crear historias y dentro de esta carrera hacer un buen videojuego es el equivalente a hacer una buena historia. Además para que sea interesante desde el punto de vista del Grado de informática propongo la generación procedural del mapa, la generación de los itinerarios o historias y la programación del comportamiento de las diversas criaturas. Sé que es un proyecto muy grande para las horas que se piden para un TFG, pero puedo dejar fuera del conteo todo el trabajo de diseño y animación y tratarlo como si fuese trabajo externo (Estaría cooperando conmigo mismo).

## Anexo B. Formulario online

Respuestas al formulario online publicado en las redes sociales.

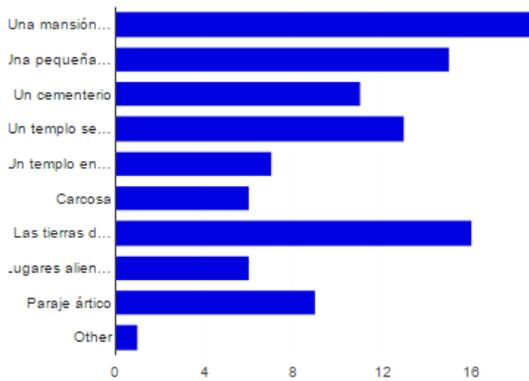
### Ambientación

Cuando imaginas jugar en el mundo de los Mitos de Cthulhu ¿En qué época te gustaría hacerlo?



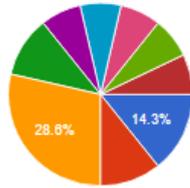
Siglo XIX	4	14.3%
Principios del siglo XX (años 20, ambientación tradicional en los juegos de rol de esta temática)	16	57.1%
Actualidad	7	25%
Futuro lejano	1	3.6%

¿Qué lugares te gustaría visitar en el juego? (Intenta votar un máximo de 4)



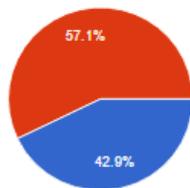
Una mansión abandonada/encantada	19	67.9%
Una pequeña localidad decadente	15	53.6%
Un cementerio	11	39.3%
Un templo sepultado en las arenas del desierto	13	46.4%
Un templo en las entrañas de la jungla	7	25%
Carcosa	6	21.4%
Las tierras del sueño	16	57.1%
Lugares alienígenas	6	21.4%
Paraje ártico	9	32.1%
Other	1	3.6%

Como protagonista serás un sectario en busca de poder, pero todo sectario tiene un pasado ¿Cual te gustaría que hubiese sido?



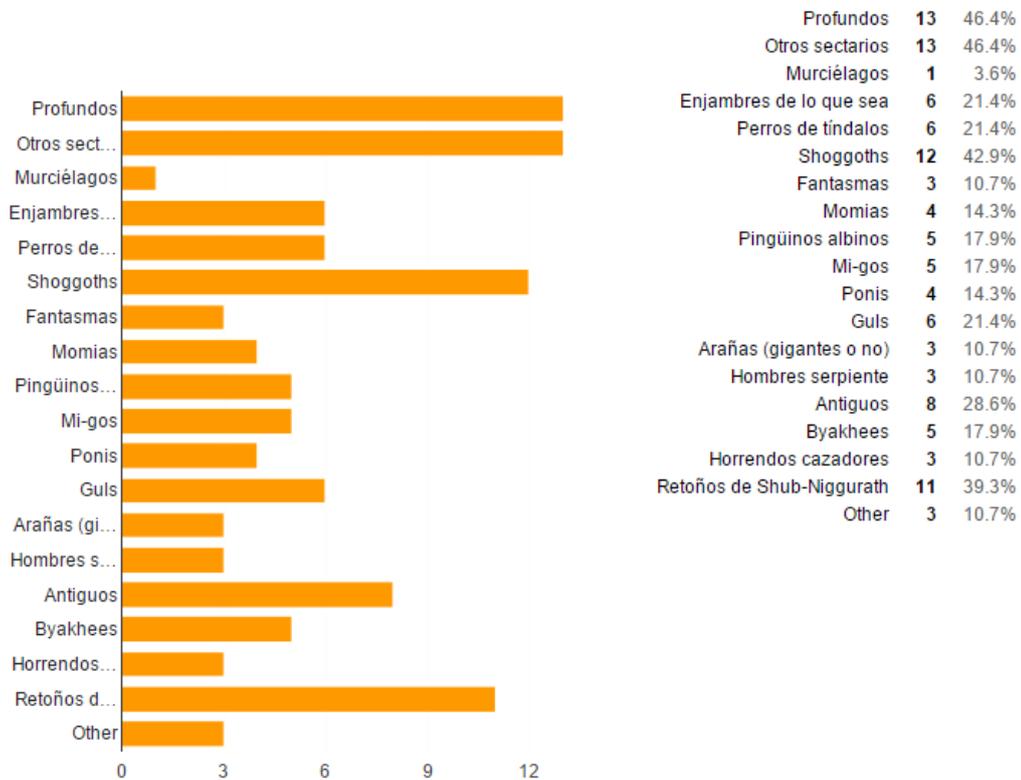
Profesor de arqueología	4	14.3%
Profesor de literatura	3	10.7%
Catedrático de geometría no euclídea	8	28.6%
Investigador privado	3	10.7%
Policía	2	7.1%
Gángster	2	7.1%
Artista	2	7.1%
Matón	2	7.1%
Other	2	7.1%

Una rapidita ¿Magia llamativa o discreta?

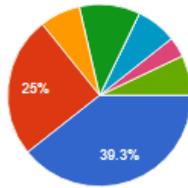


Cuanto más fuegos artificiales mejor	12	42.9%
El poder se mide por los resultados y no por el gasto en efectos especiales	16	57.1%

Si tuvieras que elegir cuatro criaturas para que aparezcan como enemigos...



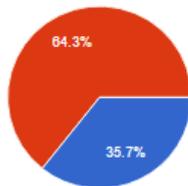
¿A quién prefieres adorar? (Tranquilo, quedará entre nosotros, ya sabemos que las deidades son celosas)



Cthulhu, Ìa Ìa	11	39.3%
Nyarlathep, el mensajero de los dioses y faraón negro.	7	25%
Azathoth ¡El dios ciego y loco!	2	7.1%
Hynos, dios de las tierras del sueño.	3	10.7%
Shub-Niggurath, la cabra de los mil retoños. Estudiosos contemporáneos afirman que cabra es un eufemismo...	0	0%
Yog-Sothoth. La llave y la puerta.	2	7.1%
¡Hastur, Hastur, Hastur! El rey de amarillo, señor de Carcosa, locos y artistas	1	3.6%
Tsathoggua. ¡Un sapo-murciélago descomunal! Es tan mono ^^	2	7.1%
Yig, padre de serpientes.	0	0%
Other	0	0%

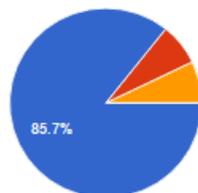
## Género

¿En un juego de plataformas cuántos saltos te gusta poder realizar antes de volver a tocar el suelo?



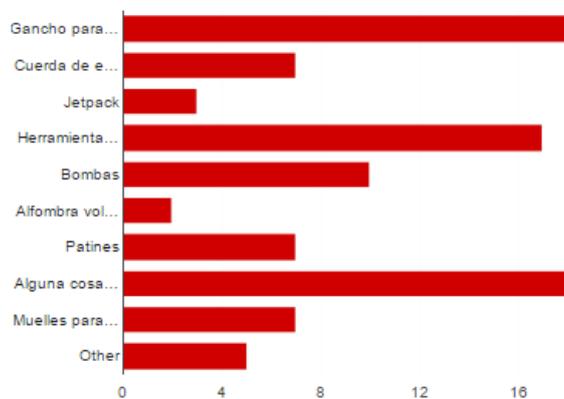
Uno, como en la vida real, saltas una vez y hasta que tus pies no toquen suelo nada de volver a saltar.	10	35.7%
Dos. Uno desde el suelo y otro en cualquier momento de la trayectoria. ¡Que esto es un juego, vamos a pasarlo bien!	18	64.3%

¿Qué método prefieres para saltar?



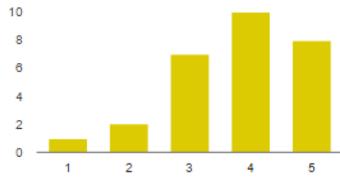
Botón de salto	24	85.7%
Tecla de flecha o joystick dirección hacia arriba	2	7.1%
Other	2	7.1%

De los siguientes, ¿qué gadgets te gusta encontrar en los juegos de plataformas? (Intenta coger un máximo de 4)



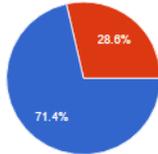
Gancho para trepar	18	64.3%
Cuerda de escalada	7	25%
Jetpack	3	10.7%
Herramientas de destrucción de escenario	17	60.7%
Bombas	10	35.7%
Alfombra voladora	2	7.1%
Patines	7	25%
Alguna cosa para planear	18	64.3%
Muelles para saltar más	7	25%
Other	5	17.9%

**Los coleccionables aportan rejugabilidad.**



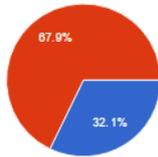
Totalmente en contra:	1	3.6%
	2	7.1%
	3	25%
	4	35.7%
Totalmente de acuerdo:	5	28.6%

**¿Qué prefieres?**



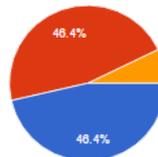
Gran variedad de armas	20	71.4%
Pocos tipos de armas	8	28.6%

**Volvamos a la magia... ¿qué opinas de su uso?**



Los primigenios proveen, así que lancemos hechizos cuando queramos.	9	32.1%
Un gran poder conlleva un gran coste. Los hechizos son algo poderoso y no pueden estar disponibles constantemente (gasto de maná o similares)	19	67.9%

**Jefes finales...**

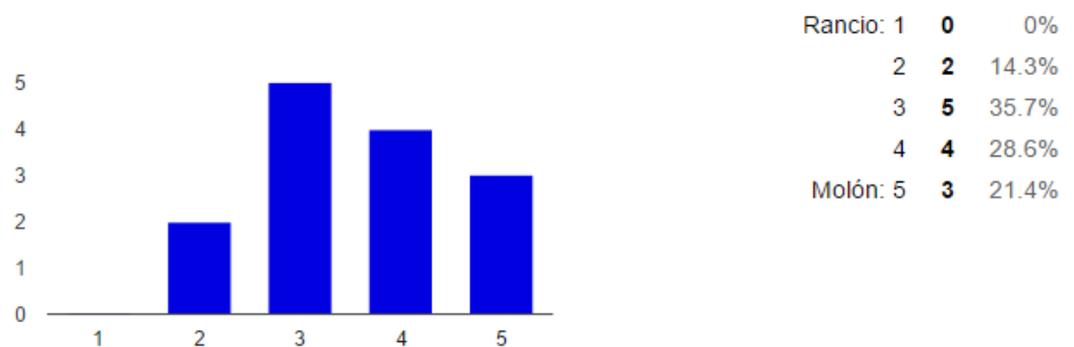


Grandes, ¡grandes! Cuanto más grandes mejor	13	46.4%
Variados	13	46.4%
De tamaño similar al jugador	2	7.1%

## Anexo C. Sesión de pruebas 2

Respuestas de los usuarios en la segunda sesión de pruebas presencial.

El estilo gráfico "definitivo" del juego, me ha parecido:



¿Tienes algo que comentar respecto al apartado gráfico? Quejas, sugerencias....

Todo guay

Los personajes de gustan, el estilo de las plataformas (en los niveles al aire libre) podría hacerse más atractivo.

Creo que el tamaño de los objetos podría cambiarse, no se si es que yo estoy medio ciega pero no los veo a veces. Añade más cosas! Quedaría más chulo si por ejemplo la mansión estuviera abarrotada de cachivaches y trastos.

Se puede mejorar la respuesta de los enemigos al ser dañados

Poca diferencia entre objetos y fondo y poca visión en la pantalla(bordes)

nada

todo bien

hay pequeños bugs visuales, como el cono de las partículas al saltar. El cono al dejarlo, cae al infinito

Le queda bastante por mejorar: animaciones, menús, minimapa, nivel de detalle en los escenarios, etc.

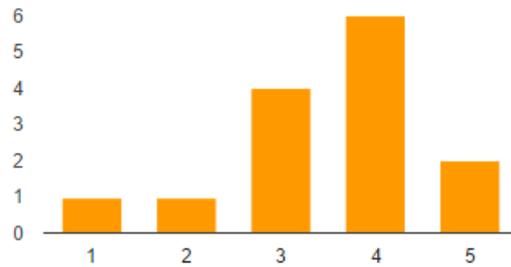
La textura de las cuerdas, please.

Más fondos, los conos de fuego requieren trabajo

estaría bien diversidad de escenarios , pero esta muy bien.

Los fondos son muy sosos. El estilo está bien.

### El control del juego me ha parecido:



Torpe: 1	1	7.1%
2	1	7.1%
3	4	28.6%
4	6	42.9%
Preciso e intuitivo: 5	2	14.3%

### ¿Tienes algún comentario respecto al control? Quejas, sugerencias

Siempre se puede pulir mas

Poder andar más rápido

Movimientos y ataque bien, pero las caídas son demasiado lentas (y la colisión del hechizo de hielo poco intuitivo).

El salto me sigue pareciendo aberrante, pero si a ti te gusta... Me parece incómodo.

Nada

El botón R queda lejos de zxc para pulsarlo sin retirar la mano. Es difícil coger pociones.

Me gustaría que el botón utilizado para entrar por una puerta no sea el mismo que de acción/atacar, puesto que a veces quieres terminar con un monstruo sin salir de la habitación.

no me gusta la barra espaciadora para saltar

al moverse el escenario el tiempo de reaccion es muy pequeño

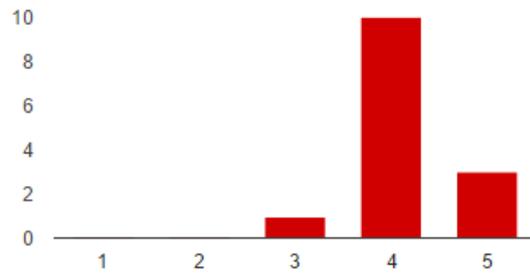
Creo que el personaje es demasiado lento respecto a los enemigos. Algunas cosas se sienten algo incómodas: interactuar con objetos, tener que saltar para sobrepasar desniveles pequeños del terreno (en lugar de que suba automáticamente)

Para mi me parece poco intuitivo, pero eso se soluciona con el poder cambiar los controles.

El doble salto no es necesario pero podría ser interesante incluirlo en circunstancias concretas los pies del byakhee estan muy op (no es una queja)

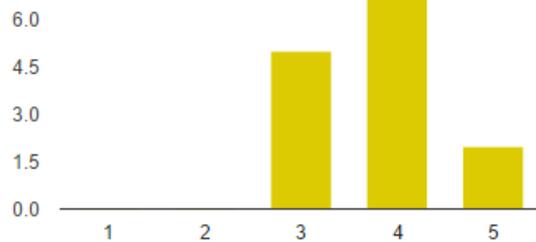
Cambiar los botones de coger y matar.

**Califica la dificultad de aprendizaje del juego (entendimiento de controles, metas y mecánicas)**



Imposible de aprender, no me aclaro:	1	0	0%
	2	0	0%
	3	1	7.1%
	4	10	71.4%
Simple e intuitivo:	5	3	21.4%

**¿Cuál ha sido tu grado de satisfacción con el juego?**



Nada satisfecho:	1	0	0%
	2	0	0%
	3	5	35.7%
	4	7	50%
Muy satisfecho:	5	2	14.3%

**¿Cómo se podría mejorar?**

Pulir el par de cosas que he puesto con anterioridad.

Nada

Es necesaria una forma de "soltar" hechizos.

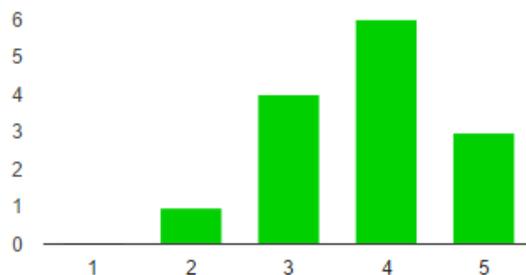
Dos veces a punto de pasarlo, dos veces que no me ha aparecido las puertas en la sala roja. El knockback es muy bajo

Barras de vida en los personajes?, textura de las cuerdas más acorde a lo que es el estilo del juego , muchas veces no sabia cuanta vida tenia por no mirar arriba. Los fantasmas creo que no deberían salirse tanto de las salas hay veces que tardan en coger agro. Ruido de daño recibido o efecto de color rojizo? {La transparencia no se distingue bien a veces}.

Necesita más armas y hechizos, el comportamiento de los enemigos resulta un poco errático a veces.

Algún objeto que de ventajas pasivas más. El salto está bien, pero quizás algo de velocidad de movimiento o más daño, molaría.

### ¿Qué tal el grado de inmersión?



Todo demasiado artificial y metido con calzador: 1	0	0%
	2	7.1%
	3	28.6%
	4	42.9%
Una buena experiencia: 5	3	21.4%

### ¿Cómo se podría mejorar?

En modo cooperativo

No es muy inmersivo, pero es debido al estilo de juego no a una mala realización.

Añadiendo más salas con más temáticas y más enemigos diferentes, aparte de otros ítems también temáticos (Por ejemplo un nivel que se desarrolle en un sótano y esté a oscuras. El personaje llevaría una luz y solo podría ver a una distancia pequeña de si mismo. Entonces el sonido serviría para avisar de la presencia de enemigos (no pongas en este nivel putos Byakhees!!!)) Añadiendo más voces y sonidos (ya te mandaré alguno)

Nada

Con una introducción e historia adecuados.

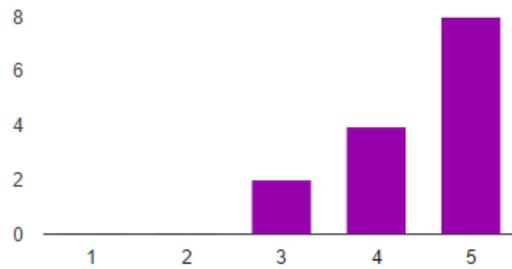
No lo se, más metas

En monitores grandes, se ve demasiado negro tanto en las salas, como en los mapas abiertos.

Simbología cthulhuoidea por las paredes y fondos, juegos de luces en situaciones concretas

Quizas intentando que haya cambios en la iluminacion? No tengo ni idea de esas cosas.

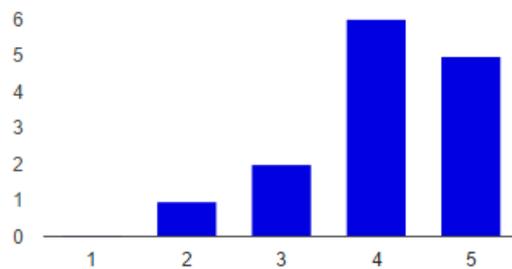
### ¿Te has sentido motivado a terminar el nivel?



No, no tenía ningún interés:	1	0	0%
	2	0	0%
	3	2	14.3%
	4	4	28.6%

Ha sido emocionante y quería ver si era capaz y qué pasaba: 5 **8** 57.1%

### ¿Y qué tal de emociones? (Diversión, tensión, cualquier estado de ánimo cuenta)



Encefalograma plano:	1	0	0%
	2	1	7.1%
	3	2	14.3%
	4	6	42.9%

Ha habido momentos memorables: 5 **5** 35.7%

### ¿Cómo se podría mejorar?

El juego es un reto que te pone en tensión, es divertido y frustrante cuando te equivocas. Es difícil, pero no imposible. Está bien hecho.

La dificultad lo hace atractivo. No obstante el spawn de los bichejos puede ser un poco "abusivo".

Que los byakhees hagan ruidos cuando se acercan hacia a ti, por favor

Nada

Añadiendo más explicación del trasfondo

Puede haber alguna fase más estresante, como con límites de tiempo, que alguna circunstancia (lava, un desprendimiento...) te obligue a recorrer salas sin poder retroceder...

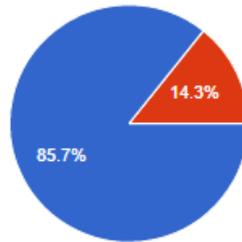
Putos aleteos

Un reto final en la sala roja para conseguir "un tesoro" y así pasarte el nivel.

Bolas aplastantes

Más variedad de enemigos y alguna trampa molarían.

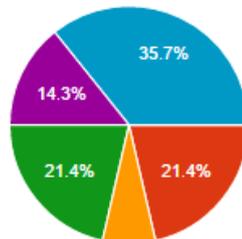
**Si te digo que falta más de un 80% del contenido del juego ¿comprarías la versión definitiva con lo que has visto hoy?**



¡Por supuesto! 12 85.7%

No, tengo muchas otras cosas mejores en las que gastar. 2 14.3%

**¿Qué precio pondrías a la versión definitiva de este juego?**



1€ 0 0%

5€ 3 21.4%

9€ 1 7.1%

10€ 3 21.4%

15€ 2 14.3%

Other 5 35.7%

**¿Alguna otra idea para mejorar el juego?**

Hacer las hitboxes de los interruptores un poco más grandes, a veces cuesta acertar para abrir las puertas. Arreglar las erratas del inicio: Ha llegado, seguro médico, cómo.

oc

Cambiar la cámara del juego, no se ve muy claro y es demasiado amplia para que sea cómodo a la vista. Cambiar toda la interfaz de usuario, es súper fea. Hacer algún tipo de menú infográfico con el daño de las armas, hechizos y enemigos y una breve descripción. A parte de añadir más.

Nada

Añadir personalización del PJ/Darle personalidad al PJ

Meter objetivos que animen a volver a jugar (desbloquear objetos, un sistema de logros...) En la prueba he encontrado un error: al cruzar una puerta agarrado a una cuerda apareces en la siguiente sala agarrado a una cuerda invisible.

más hechizos y monstruos

Más cosas y modo dios

- Aumentar el rango interacción de las pociones, etc. - Hacer que la cámara enfoque el menor trozo negro posible - Hacer que los objetos se cojan automáticamente y que al soltarlos salgan despedidos - Mejorar la visibilidad de las dagas

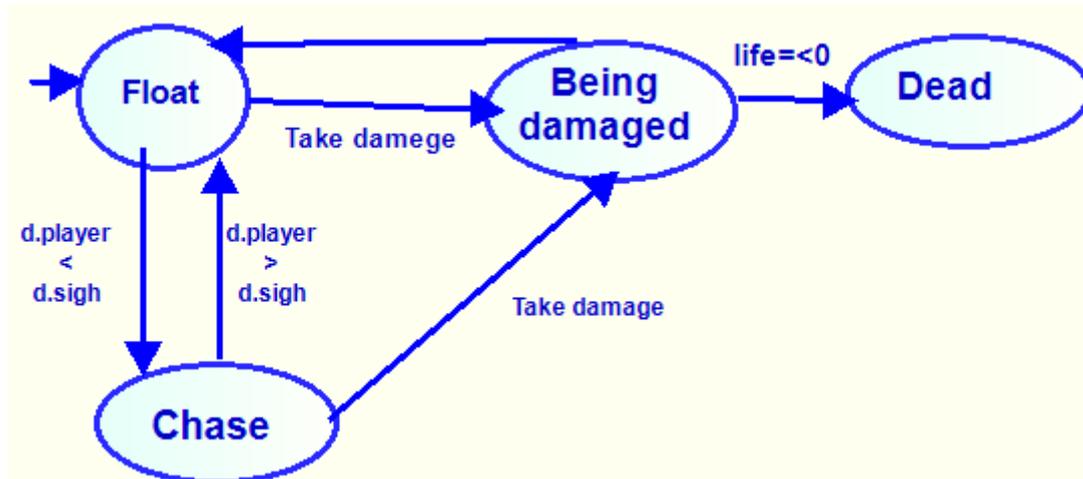
Un reto final en la sala roja para conseguir "un tesoro" y así pasarte el nivel, poder seleccionar el nivel de dificultad entre varios uno de ellos experto en plan 1 toque mueres, En el minimapa poner colores de distintas gamas (Más diferenciados), xdd yo el verde y el amarillo no los distinguía bien.

Presencia explícita de primigenios y dioses exteriores.

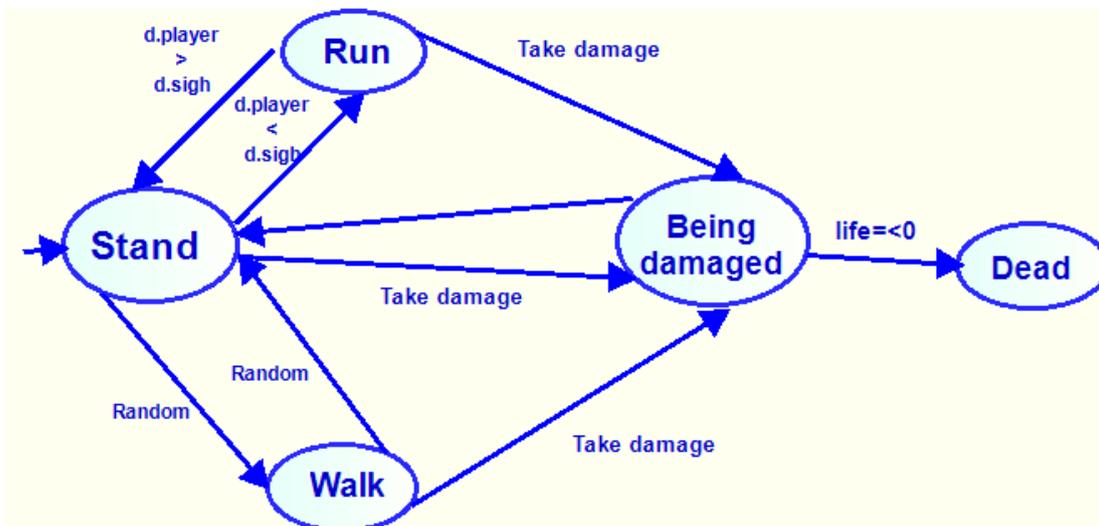
## Anexo D. Diagramas de estado de enemigos

A continuación se muestran los diagramas de estado de los tres enemigos implementados para la prueba.

### Catherine



### Deeper



Byakhee

