

Catedral - Laberinto de Axiom

Kevin Esguerra Cardona

Concepto

- Partidas rápidas, caóticas.
- Roguelike.
- Progresión de poder en función del tiempo.
- Cada partida debe sentirse única.
- La emoción prima sobre la narrativa.

Trasfondo

Un estudiante universitario muere de cansancio mientras desarrolla un videojuego y despierta en un laberinto en tinieblas. De repente, una voz le dice que es un héroe convocado para acabar con el Custodio del Tiempo y detener el reloj del caos antes de que su tiempo se agote y una entidad más allá de la comprensión sea liberada. Los diálogos y cuadros de personajes se describen en el anexo 1: trasfondo.

Clases jugables

El jugador tendrá a su disposición 5 tipos de héroes distintos: caballero, guerrero, paladín, mago y arquero. Estas clases y sus atributos se describen en el anexo 2: clases jugables.

Sistema de magia

En el juego se implementará un sistema de magia basado en escuelas: elemental, mística, somática y entrópica. El sistema de magia se comparte entre el jugador y los enemigos. Estas

escuelas y cada uno de los hechizos que contienen se describen en el anexo 3: sistema de magia.

Objetos

El juego implementará un sistema de objetos con progresión dinámica. En el mapa del juego, de manera pseudo-aleatoria aparecerán objetos que el jugador podrá recoger. Estos objetos son de dos tipos: portables, armas y armaduras; y usables, pócimas. La calidad de los objetos generados deberá aumentar en función del tiempo, es decir, a partidas más largas, objetos más valiosos. Las reglas de generación y calidades de los objetos se describen en el anexo 4: objetos y su sistema de progresión.

Enemigos

El jugador podrá encontrar 5 clases de enemigos. Goblin, chamán, caballero oscuro, señor de la guerra y el Custodio del Caos. Su jerarquía y habilidades se describen en el anexo 5: enemigos.

Mapa

El mapa del juego se generará procedimentalmente, evitando una sensación de fatiga al jugar una y otra vez en el mismo mapa. Inicialmente, se prevé un mapa de 64 por 64 casillas de 48 píxeles cada una. Usando una variación del algoritmo Prim se generará un laberinto en la cuadrícula de 64 casillas con aproximadamente un 30% de ramificaciones. El 10% de las casillas de suelo se sellan para colocar trampas. La sala del reloj del caos se encontrará siempre a no menos de 40 casillas (medida Manhattan) del punto de aparición del jugador. El algoritmo específico de generación del mapa se describe en el anexo 6: mapa.

Estilo e inspiración

El juego tendrá una estética gótico-futurista con aires de steampunk en los enemigos de alto nivel. La ambientación debe ser atrapante y opresiva. El arte será pixelart y la ambientación musical y sonora seguirá las mismas afluentes de los gráficos. El estado del arte y los ejemplos y bocetos en este apartado se describen en el anexo 7: estilo e inspiración.

Arquitectura

Como motor de desarrollo se optó por PyGame, con posibilidades de integrar herramientas más potentes en caso de que el proyecto escale fuera de un ambiente meramente académico. Los datos estáticos se almacenarán en JSON o YAML para una alta disponibilidad en términos de accesibilidad y modificabilidad. Se decidió usar el patrón de desarrollo basado en componentes con mixins para agilizar los cálculos y la toma de decisiones en tiempo de ejecución, esto se debe a la alta demanda en procesamiento que exige el género Roguelike. Los enemigos implementarán distintas IAs en función de la dificultad que deban representar para el jugador. Los detalles sobre la arquitectura, esquemas, diagramas de clases, secuencia y flujo; así como las IAs implementadas por cada tipo de enemigos se describen en el anexo 8: arquitectura.

Plan de desarrollo

Hito 1 - Semana 1

GDD → prototipo → grid/base del mapa → movimiento → ataques básicos.

Para la primera semana se espera completar el documento de diseño junto con todos sus anexos, así como el primer prototipo del juego implementando la base del mapa del juego, es decir, una grid o cuadrícula de 64 por 64 casillas, el movimiento del PJ y sus ataques básicos.

Hito 2 - Semana 2

Prim → clases

Para la segunda semana se espera implementar el algoritmo de generación de laberintos Prim, así como las distintas clases de PJ con sus respectivas habilidades y estadísticas.

Hito 3 - Semana 3

Loot → UI mínima → assets

Para la tercera semana se espera implementar el sistema de generación de loot y la interfaz mínima para ingresar al juego, configurar e interactuar (iniciar, jugar y terminar una partida).

También se debe implementar assets para los objetos principales del juego (muros, suelo, PJ, PNJ, Loot)

Hito 4 - Semana 4

IA → Sistema de magia

Para la semana 4 se espera implementar la IA para los enemigos y el sistema de magia completo.

Hito 5 - Semana 5

Balance de parámetros → sonido → detalles visuales

Para la quinta semana se espera realizar un balance inicial de los parámetros de dificultad del juego, así como el sonido (ambientación y banda musical) y los detalles visuales (luces, sombras, partículas, assets faltantes).

Hito 6 - Semana 6

Pulido → empaquetado

Para la sexta semana se espera realizar un pulido final del juego, su interfaz y su documentación. También se debe empaquetar (ejecutable) y generar la primera versión jugable (v0.1).