

Historia

En un futuro no muy lejano, la humanidad ha alcanzado un avance tecnológico sin precedentes. Sin embargo, este progreso ha traído consigo una crisis de identidad y caos. Las emociones humanas han sido reemplazadas por la lógica fría de las máquinas, y la esencia de lo que significa ser humano ha comenzado a desvanecerse. Para recuperar lo que se ha perdido, un grupo de científicos ha creado un experimento radical: las Cápsulas, entidades biológicas que representan la dualidad del ser humano.

Cada cápsula estaba compuesta por dos entidades que fueron separadas por meiosis: el Estratega y el Guerrero. El Estratega es capaz de ver más allá de las limitaciones del laberinto, detectando trampas y obstáculos, mientras que el Guerrero posee habilidades ofensivas para eliminar cualquier amenaza que se interponga en su camino. Juntos, deben navegar a través de un laberinto cambiante, lleno de peligros y sorpresas, donde la única forma de sobrevivir es colaborar y usar sus habilidades de manera inteligente.

El laberinto fue diseñado por un antiguo grupo de filósofos que creían que el verdadero sentido de la vida reside en la lucha por el autoconocimiento y la superación. En el centro del laberinto se encuentra un objeto sagrado conocido como el Corazón de la Humanidad, un artefacto que contiene la esencia de las emociones perdidas. El objetivo del juego es llegar al Corazón antes que los demás jugadores, restaurando así la conexión con lo que significa ser humano.

Descripción del juego

Encapsulados es un juego multijugador por turnos que tiene como objetivo llegar hasta la ubicación de un objeto dado, superando tanto las trampas del laberinto como las amenazas de otros jugadores.

Género:

Estrategia multijugador (PvP).

Tecnologías utilizadas:

C#: Lenguaje de programación utilizado.

Unity: Motor de juego utilizado para la interfaz visual.

Git: Herramienta de control de versiones.

Mecánica del juego:

Objetivo: Llevar a los 2 personajes hasta el objeto señalado en el mapa.

Turnos: En cada turno un jugador puede seleccionar uno de sus personajes, explorar el laberinto y usar su habilidad mientras no termine su turno (el turno de cada personaje tiene un tiempo limitado).

Habilidades: Los **ESTRATEGAS** son pensadores por naturaleza por lo que sus habilidades suelen ser bastante estratégicas y nunca pueden atacar. Mientras tanto los **GUERREROS** están ahí precisamente para defender (para ellos un ataque es la mejor forma de defensa) por lo que sus habilidades pueden dañar al enemigo..

Obstáculos y trampas: Mientras recorren el laberinto los jugadores pueden caer en trampas (una trampa es visible para un personaje solo si ya ha caído en esta antes o alguna habilidad le permite ver esta trampa), pueden encontrarse con puertas (una puerta solo se puede abrir con la gema adecuada para esta o con una habilidad que lo permita) y por último pueden encontrar también las gemas antes mencionadas.

Laberinto dinámico: Al iniciar cada partida el laberinto es generado aleatoriamente al igual que la posición del jugador. El jugador puede ver en el mapa la ubicación del su objetivo, sin embargo no puede ver su propia posición. Para descubrir su posición y lograr llegar a su objetivo tendrán que intentar ubicarse buscando en el mapa los caminos mas parecidos a los que ha recorrido.

Interfaz de usuario:

El juego cuenta con una interfaz que permite visualizar el laberinto a través de un mapa y moverse por el laberinto fácilmente.

Estructura del Proyecto

El proyecto está dividido en:

1. Clases de la lógica de generación del laberinto:
 - Maze.cs: Encargado del algoritmo de generación del laberinto.
 - Cell.cs: Clase para las propiedades de las celdas del laberinto.
2. Clases para controlar el estado del juego:
 - Menu.cs: Controlador del menú principal del juego.
 - CharacterMenu.cs: Controlador del menú de selección de los personajes.

- Container.cs: Se encarga de instanciar o modificar todos los objetos que se puedan ver en la escena. Para tener el laberinto en escena utiliza las clases de generación del laberinto.
 - GameManager.cs: Controlador de la escena principal del juego. Se encarga de la inicialización, pausa y control de eventos del juego durante una partida.
 - Global.cs: Contiene variables accesibles desde todas las clases.
3. Clases para controlar los personajes:
 - MovPlayer.cs: Controla el movimiento de los jugadores.
 - Character.cs: Contiene las propiedades de cada personaje (vida, habilidad, duración de su turno, etc) y controla algunas acciones del personaje como el uso de las habilidades.
 4. Clases para controlar los objetos del juego:
 - Obstacle.cs
 - Trap.cs
 - Keys.cs
 - Final.cs

Todos estos controlan las interacciones de los jugadores con los mismos objetos.

Ejecución del proyecto

Requisitos:

Unity 6000.0.25.f1

Pasos para ejecutar el proyecto en Unity:

1. Clonar el repositorio de GitHub. Para esto puede utilizar el siguiente comando en la terminal: `git clone https://github.com/Edo-06/MazeUnity.git`
2. Navega hasta la carpeta raíz del proyecto.
3. Abre Unity Hub, seleccione la opción "Abrir proyecto" y busque la carpeta Maze en el repositorio clonado.
4. Unity abrirá el juego en el editor, corra el proyecto pulsando el botón play.

Pasos para abrir el juego directamente:

1. Clonar el repositorio de GitHub. Para esto puede utilizar el siguiente comando en la terminal: `git clone https://github.com/Edo-06/MazeUnity.git`
2. Navega hasta la carpeta raíz del proyecto.
3. Abre la carpeta "Ejecutable" y descomprima "Encapsulado".

4. Una vez realizados estos pasos al entra a la carpeta "Encapsulado" podrá abrir el .exe y jugar.

Instrucciones para jugar:

W: Moverse hacia adelante

S: Moverse hacia atrás

A: Moverse hacia la izquierda

D: Moverse hacia la derecha

Q: Para usar tu habilidad