

Instituto Tecnológico de Costa Rica

Curso: Introducción a la Programación

Proyecto 2 – Escapa del Laberinto

Semestre II, 2025

Estudiantes: Christian David Esquivel Álvarez

Alejandro Salazar Barboza

Grupo: 1

Documentación del Proyecto – Escapa del Laberinto

1. Atributo de Análisis de Problema

El problema complejo de ingeniería, consiste en diseñar un sistema interactivo donde el jugador debe escapar de un laberinto generado, evitando a los cazadores que funcionan con algoritmos de persecución y comportamiento autónomo. Este sistema requiere la integración de principios matemáticos, computacionales y diseño.

El mapa se modela como una matriz numérica que funciona como un grafo, donde cada casilla representa un tipo de terreno con restricciones específicas. El jugador y los cazadores deben interactuar respetando las reglas del entorno.

Análisis del contexto

Las principales variables del sistema incluyen terreno, energía del jugador, velocidad y número de cazadores, lógica de persecución, trampas con cooldown, generación aleatoria del mapa y tiempos de respawn. Estas variables deben interactuar sin generar errores de colisión o rutas inválidas dentro del laberinto.

Plan de solución

La solución se implementa mediante un modelo modular compuesto por clases: Jugador, Enemigo, Trampa y casillas de terreno. Se utilizan técnicas como BFS para validar la existencia de caminos viables y para la persecución enemiga. El sistema de trampas emplea timers y control de eventos, mientras que el puntaje combina tiempo, dificultad y enemigos eliminados.

Evaluación de soluciones (Pros y Contras)

Mapa procedural: Gran jugabilidad, pero puede generar rutas estrechas.

Trampas temporizadas: Aportan estrategia, pero requieren sincronización precisa.

Modelo energético: Añade balance, pero requiere calibración de parámetros.

Respawn de enemigos: Mantiene la dificultad, pero debe evitar posiciones cercanas al jugador.

2. Atributo de Herramientas de Ingeniería

Para resolver el problema se utilizan técnicas de programación orientada a objetos, manejo de matrices, teoría de grafos, control de tiempo con temporizadores y el motor de eventos de Pygame. Estas herramientas permiten modelar comportamientos de entidades y gestionar la lógica del juego de forma clara y eficiente.

Aplicación de las herramientas durante el proyecto

La POO se usa para encapsular comportamiento en clases.

Pygame gestiona el bucle principal, eventos, HUD y colisiones.

Los timers controlan energía, cooldown de trampas y respawn.

La generación aleatoria produce mapas únicos en cada partida.

Adaptación de herramientas

Durante el desarrollo del proyecto, fue necesario ajustar valores, modificar la velocidad de los enemigos según la dificultad, optimizar el BFS y garantizar que el mapa generado fuese siempre jugable. También se modificaron los controles y la visualización del HUD para mejorar la experiencia del usuario.

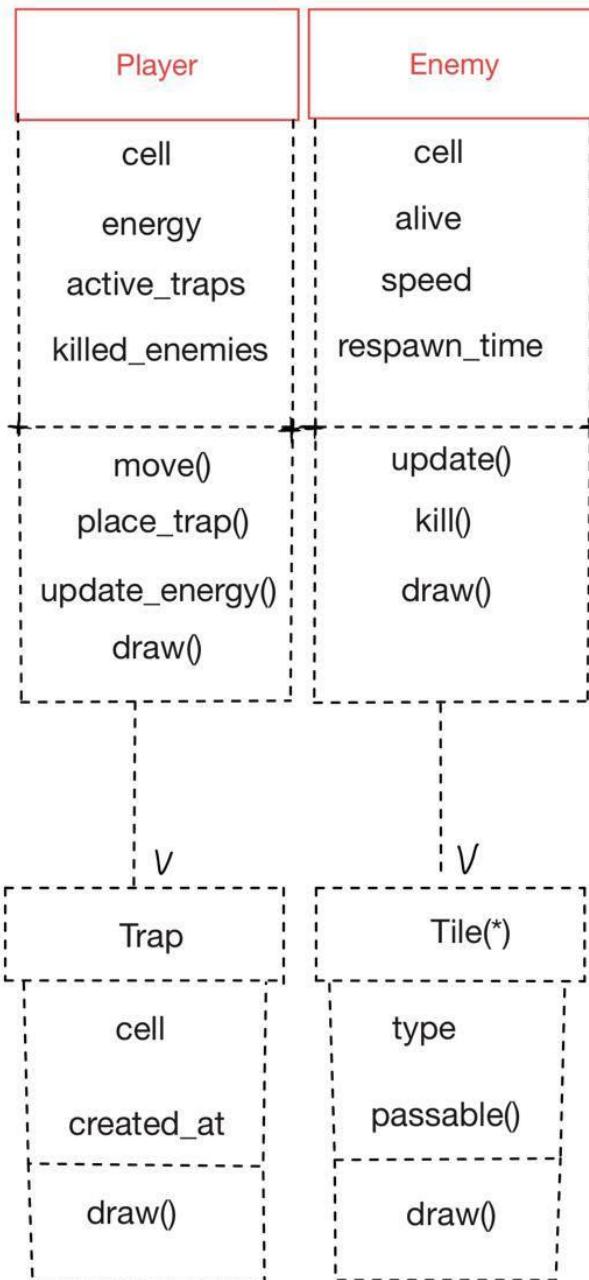
Diagrama de Clases

Player: Movimiento, energía y trampas.

Enemigo: Persecución y respawn.

Trap: Activación con cooldown, eliminación de enemigos.

Tile: Representación de casillas del mapa.



Modo Cazador

En este modo, el jugador toma el rol de cazador y debe atrapar enemigos antes de que alcancen la salida. La IA cambia completamente: los enemigos ya no persiguen al jugador, sino que huyen de él, buscando rutas de escape. Cada enemigo atrapado suma puntos y cada enemigo que escapa resta puntuación. Este modo utiliza rutas inversas con BFS y algoritmos aleatorios para simular evasión.