

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computadores

Curso: Introducción a la Programación, Grupo 2

Profesor: Jeff Schmidt Peralta

Documentación del Proyecto: Escapa del laberinto

Daniel Araya e Isaac Orozco

II Semestre 2025

1. Atributo de Análisis de Problema

Desde el inicio, mi compañero Isaac y yo nos propusimos el reto de desarrollar un videojuego educativo que simulara la dinámica de escape y persecución en un laberinto generado aleatoriamente. El problema principal que enfrentamos fue diseñar un sistema capaz de representar un entorno dinámico, donde tanto el jugador como los enemigos interactúan bajo reglas claras y restricciones de movimiento, garantizando siempre la jugabilidad y el balance.

Para abordar este desafío, recurrimos a principios matemáticos, como el uso de matrices para modelar el mapa y algoritmos de generación aleatoria que aseguran la existencia de al menos un camino válido en cada partida. Además, aplicamos conceptos de ciencias naturales al simular la energía del jugador, lo que añade realismo y estrategia al juego. Desde la perspectiva de la ingeniería, nos enfocamos en la modularidad y la reutilización de código, lo que nos permitió mantener el proyecto organizado y facilitar futuras mejoras.

El contexto educativo de este proyecto nos motivó a priorizar el aprendizaje y la colaboración. Consideramos variables como la dificultad, el balance entre los modos de juego, la gestión de la energía y la inteligencia artificial básica para los enemigos. Todo esto lo integramos pensando en el desarrollo sostenible, ya que buscamos crear una base sólida y flexible que pueda ser reutilizada o ampliada en el futuro.

Nuestro plan de solución consistió en modelar cada tipo de casilla como una clase independiente, implementar una clase para los enemigos y otra para el jugador, y sincronizar cuidadosamente los eventos del juego (movimiento, trampas, energía) para evitar errores y garantizar una experiencia fluida. Evaluamos constantemente nuestras decisiones, identificando ventajas como la claridad del código y la facilidad de mantenimiento, y desventajas como la complejidad de la sincronización y el ajuste de la dificultad. Sin embargo, creemos que el enfoque modular y sostenible que adoptamos es el más adecuado para este tipo de proyectos.

Pros:

- Modularidad y claridad en el código.
- Facilidad de mantenimiento y expansión.

- Aprendizaje práctico y trabajo colaborativo.

Contras:

- Complejidad en la sincronización de eventos.
- Ajuste fino de la dificultad y la IA.
- Posibles errores en la generación aleatoria.

Aun así, el enfoque modular y sostenible que adoptamos es el más adecuado para este tipo de proyectos, permitiendo su mejora continua y adaptación a nuevas necesidades.

2. Atributo de Herramientas de Ingeniería

Para resolver el problema, utilicé principalmente la programación orientada a objetos, lo que me permitió representar de manera clara y ordenada los distintos elementos del juego: el mapa, los tipos de casillas, el jugador, los enemigos y las trampas. Implementé algoritmos de generación de laberintos, asegurando que cada partida fuera única y siempre jugable. La matriz numérica fue fundamental para gestionar el entorno y las restricciones de movimiento.

Además, desarrollé una interfaz gráfica intuitiva, que facilita la interacción del usuario y la visualización de la energía y las trampas. El control de versiones con Git y el uso de un repositorio en GitHub fueron esenciales para coordinar el trabajo con Isaac, registrar avances y mantener un historial claro de cambios.

Durante el desarrollo, apliqué y adapté diversas técnicas según las necesidades del proyecto. Por ejemplo, ajusté los algoritmos de generación de mapas para mejorar la jugabilidad y modifiqué la lógica de los enemigos para equilibrar la dificultad. También optimicé la gestión de eventos para evitar errores de sincronización y asegurar que la experiencia de juego fuera fluida y entretenida.

Diagrama de Clases

3. Bitácora de Trabajo

Durante el desarrollo del proyecto, Isaac y yo trabajamos de manera colaborativa, dividiendo las tareas según nuestras fortalezas e intereses, y asegurando un avance constante a través de commits frecuentes y descriptivos en nuestro repositorio de GitHub.

Aportes de Isaac:

Isaac se encargó principalmente de diseñar y programar el sistema de puntajes, asegurando que cada modo de juego tuviera reglas claras y justas para la asignación de puntos. Además, fue responsable de implementar las mecánicas principales del juego, como la

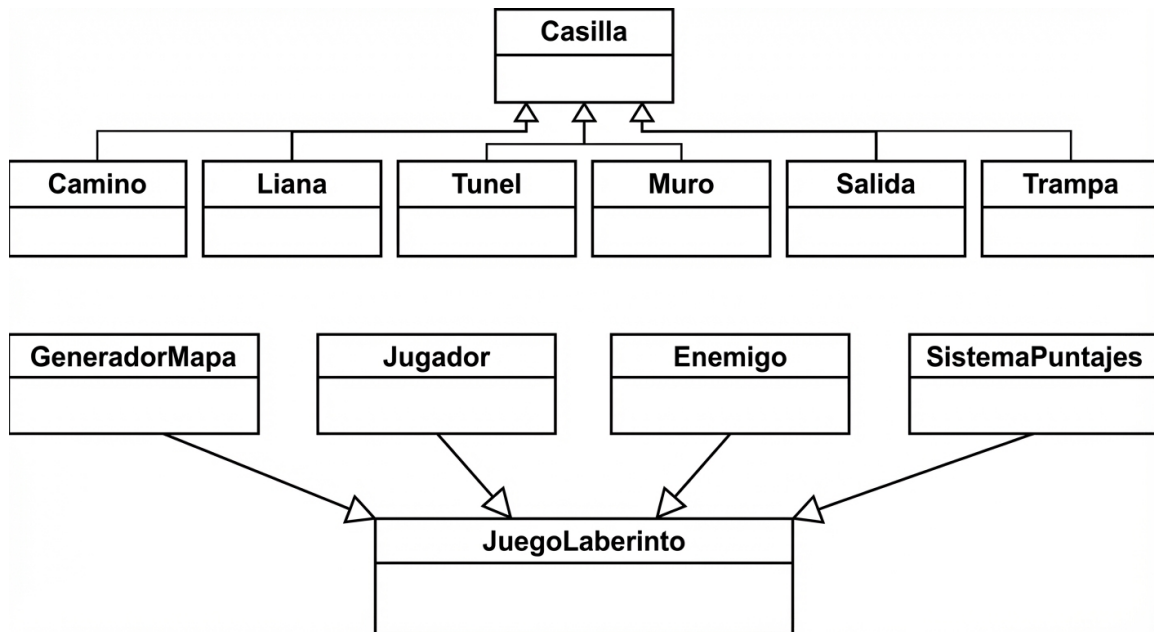


Figura 1: Diagrama de clases principal del proyecto Escapa del laberinto.

gestión de trampas, la barra de energía y la lógica de victoria y derrota en ambos modos. Gracias a su trabajo, el juego cuenta con una estructura sólida y equilibrada, que motiva al jugador a mejorar su desempeño.

Aportes míos:

Por mi parte, me enfoqué en la resolución de bugs que surgieron durante las pruebas, optimizando el código para evitar errores y mejorar la estabilidad general del juego. También me encargué de añadir los sprites y las imágenes, lo que permitió darle una identidad visual atractiva y coherente al proyecto. Finalmente, dediqué tiempo a optimizar el sistema de inteligencia artificial de los enemigos, logrando que su comportamiento fuera más desafiante y realista, lo que incrementó la calidad de la experiencia de juego.

Trabajo colaborativo y control de versiones:

Ambos documentamos nuestro progreso mediante commits frecuentes y detallados en GitHub. Cada commit refleja claramente la tarea realizada, ya sea la implementación de una nueva funcionalidad, la corrección de un error, la mejora de la IA o la actualización de recursos visuales. Esta práctica no solo facilitó la colaboración, sino que también permitió mantener un historial transparente y ordenado del desarrollo.