

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Introducción a la programación

Proyecto 2 – Escapa del laberinto

Estudiantes:

Nayelith Sojo Mendez
2025080083

Génesis Elizabeth Ayala Pineda
2025800545

Profesores:

Diego Andres Mora Rojas
Jeff Schmidt Peralta

Grupo:

4

Año:

2025

1. Atributo del Análisis Problemático

El proyecto conlleva el diseño y posterior implementación de un sistema interactivo, fundamentado en la representación laberíntica, materializada mediante una matriz. El participante se verá obligado a navegar por el entorno, administrando simultáneamente recursos finitos como la energía, y adoptando decisiones tácticas en respuesta a las variantes del terreno, además de lidiar con agentes hostiles dotados de comportamientos predefinidos. Este paradigma recrea una problemática ingenieril de alta complejidad, entrelazando variables dinámicas, restricciones espaciales y la necesidad de tomar decisiones bajo incertidumbre.

El análisis del problema emerge de la identificación del entorno como un sistema estructurado, precisamente una matriz de FILAS x COLUMNAS, que alberga diversas tipologías de terreno (camino, muro, túnel, liana y salida). Cada cual ostenta peculiaridades inherentes que impactan en la locomoción del jugador y/o de sus perseguidores. Las variables esenciales del sistema se circunscriben a la ubicación del jugador y los antagonistas, la reserva energética, el lapso temporal, la cuantía de perseguidores, el nivel de dificultad y finalmente los eventos dinámicos (como la colocación de trampas, las capturas, colisiones o fugas).

Desde la óptica del análisis contextual el problema se aborda examinando los posibles estados del sistema y las derivaciones de cada acción. A modo ilustrativo, el acto de correr incrementa la velocidad, claro, pero merma la energía; los túneles proveen una ventaja táctica singular; las lianas son útiles para los cazadores; las trampas neutralizan a los enemigos, pero precisan un cooldown. Este abordaje se vincula con principios propios de la ingeniería, entre los cuales figuran la identificación de restricciones, el uso de recursos finitos, las rutas óptimas, la eficiencia en el movimiento y la predicción del comportamiento de los agentes externos, los enemigos.

La propuesta resolutive implica la instrumentación de un modelo fundamentado en la programación orientada a objetos donde cada componente del entorno se simboliza con clases particulares y los cambios de estado son administrados a través de métodos bien delimitados. El sistema incorpora dos modos operativos: el Modo Escapa que prioriza la evasión estratégica y el Modo Cazador que se centra en el control territorial mediante la captura de enemigos. La lógica interna del juego se apoya en reglas inequívocas que rigen el movimiento, las colisiones, la energía, la puntuación y los criterios de victoria o derrota.

Finalmente, el análisis del modelo arroja que éste cumple con lo previsto; El personaje puede recorrer el escenario, los oponentes reaccionan de manera cambiante, la dificultad altera el comportamiento del sistema y las decisiones del jugador presentan consecuencias medibles. Las virtudes clave del diseño resaltan su modularidad, la nitidez en la representación del entorno, y la capacidad de expansión. Aunque existen restricciones, como la rudimentaria IA enemiga y una elevada dependencia de la interfaz visual, el sistema logra un equilibrio apropiado entre complejidad y funcionamiento, ilustrando un problema interactivo de ingeniería con acierto.

2. Atributos de Herramientas de Ingeniería

En la construcción del proyecto se implementaron diversas herramientas y técnicas esenciales de la ingeniería del software. La técnica primordialmente empleada fue la programación orientada a objetos POO, la cual facilitó la modelización del laberinto y sus componentes como entes autónomos. Se establecieron clases tales como Mapa, Casilla, Terreno, Jugador, Cazador, Trampa, y Dificultad, cada cual dotada con sus propios atributos y comportamientos encapsulados. Esta arquitectura propicia la modularidad, la reusabilidad del código y la transparencia en la visualización del sistema.

Entre los instrumentos utilizados, destaca Python, cual lenguaje central, empleado junto a la librería Tkinter, crucial en la configuración de la interfaz visual, gestionando mapas, el canvas y los sucesos de teclado. La preservación de datos se realizó con JSON, archivando las puntuaciones de los jugadores, organizadas. Adicionalmente, se incluyó winsound, una biblioteca que añade sonidos simples, optimizando la experiencia del usuario. Igualmente, GitHub fungió como sistema de control de versiones, favoreciendo un trabajo colectivo estructurado por ramas separadas.

Las tácticas aplicadas abarcaron: manipulación de matrices, control de eventos del teclado, la identificación de colisiones, la refrescante actualización cíclica del estado del sistema usando ciclos "after", administración energética, temporizadores, y el cálculo de la puntuación en función de variables dinámicas del juego. En el diseño, se desarrollaron algoritmos sencillos, determinando el desplazamiento de los oponentes, la colocación de trampas, la definición de rutas permisibles dentro del laberinto, así como el reconocimiento de estados finales ganadores o perdedores.

A lo largo del proyecto, la flexibilidad fue crucial al tener que reajustar instrumentos y métodos en respuesta a requerimientos emergentes. Consideremos el Modo Cazador; su inclusión obligó a reescribir el comportamiento de los antagonistas para evadir al jugador, además de calibrar la lógica de puntuaciones y colisiones. La aplicación del registro de jugador, a su vez, solicitó la integración de una ventana inicial; mientras el sistema de puntuaciones históricas demandó añadir la persistencia de datos y cerciorar la compatibilidad del JSON entre los distintos modos de juego. Adicionalmente, la interfaz gráfica fue perfeccionada incorporando efectos visuales y tonalidades más nítidas, para favorecer la experiencia del usuario.