



Tecnológico de Costa Rica

Curso:

Introducción a la Programación

Escuela de Ingeniería en Computadores

Profesor:

Jeff Schmidt Peralta

Integrantes:

Rolando Tula 2025115428

Carlos Zhou Fernández 2025077261

Título:

Proyecto Segundo Semestre Juego del Laberinto

27 de Noviembre

2025

Atributo de Análisis de Problema

El problema de ingeniería que se buscó resolver fue el desarrollo de un videojuego compuesto por dos modos principales. El primero, “**Modo Escape**”, consiste en que el jugador debe atravesar un laberinto y alcanzar la meta evitando a varios cazadores que lo persiguen dinámicamente. Este modo exige al usuario tomar decisiones rápidas, anticipar movimientos enemigos y planificar rutas eficientes. El segundo, “**Modo Cazador**”, invierte el rol y convierte al jugador en quien persigue a los enemigos, lo que obliga al sistema a manejar comportamientos de huida y patrones distintos para la inteligencia artificial del juego. Esta dualidad representa un desafío adicional, ya que requiere diseñar mecánicas opuestas dentro de la misma estructura base.

Además del gameplay, el proyecto incluye un **registro de usuarios y sus puntajes**, lo cual añade una capa de gestión de datos persistentes. Esto implica diseñar un sistema capaz de guardar información de forma estable, cargarla sin errores y mantenerla accesible al cambiar de menú o cerrar la aplicación.

Para resolver este problema complejo se aplicaron conceptos fundamentales: **variables, funciones, listas, condicionales, iteración, eventos**, y de manera destacada, la **Programación Orientada a Objetos (POO)**. La POO permite organizar el código en clases como *Jugador*, *Enemigo*, *Mapa*, *Botones* y *SistemaDeUsuarios*, facilitando la modularidad y la legibilidad. Esto hizo que la implementación del juego fuera más escalable y que cada componente pudiera ampliarse sin afectar a los demás.

La iteración también fue clave, ya que el juego depende de un ciclo principal que actualiza posiciones, detecta colisiones, gestiona la lógica de persecución y renderiza los elementos en pantalla de forma constante. Gracias a las técnicas estudiadas, fue posible integrar estos procesos sin afectar el rendimiento general del programa.

Atributo de Herramientas de Ingeniería

- **Python:** Se utilizó como lenguaje principal gracias a su sintaxis clara y su versatilidad. Permitió desarrollar tanto la lógica del juego como la interfaz gráfica.
- **Pygame:** Sirvió para manejar gráficos, sonidos, eventos del teclado y el ciclo principal del juego. Brindó las herramientas necesarias para construir un entorno 2D funcional.
- **Git:** Fue fundamental para controlar el historial del proyecto, revertir errores y mantener un desarrollo ordenado.

- **GitHub:** Permitió almacenar y compartir el repositorio, facilitando el trabajo colaborativo.
- **JSON:** Se empleó como método de almacenamiento local para usuarios y puntajes, debido a su estructura ligera y fácil interpretación.

Técnicas aplicadas

Reutilización de Componentes (POO):

La Programación Orientada a Objetos permite crear clases reutilizables con comportamientos definidos. Esto facilitó organizar el proyecto en módulos claros y expandir el juego sin duplicar código. Por ejemplo, los enemigos comparten métodos de movimiento, pero cada uno puede ajustarse con valores diferentes como velocidad o comportamiento.

Persistencia de Datos:

Se implementó lectura y escritura de archivos JSON para almacenar puntajes, usuarios y configuraciones. Gracias a esto, la información permanece guardada incluso si el juego se cierra, mejorando la experiencia del jugador y permitiendo un sistema de clasificación sencillo.

Control de Versiones:

Mediante Git y GitHub fue posible dividir tareas, integrar cambios sin conflictos y mantener un registro del progreso del proyecto. Esto simuló prácticas reales de la industria del software y permitió trabajar de forma organizada.

Adaptación

Python y Pygame:

El diseño general del juego, sus menús, botones interactivos, animaciones y dificultad progresiva fueron ajustados mediante estas herramientas. Se buscó que el jugador experimentará una curva de reto equilibrada y un ambiente visual claro y comprensible.

JSON:

Su uso permite almacenar datos de forma estructurada, legible y fácilmente ampliable. Esto asegura que en el futuro puedan agregarse más atributos o sistemas (como logros, tiempos récord o configuraciones personalizadas) sin modificar la arquitectura básica del almacenamiento.

