Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computadores

Curso: Introducción a la Programación

Proyecto 2 – Escapa del laberinto

Semestre II, 2025

Descripción general del proyecto

El proyecto consiste en la implementación de dos modos de juego principales, ambos desarrollados sobre un mapa representado mediante una matriz numérica. Cada casilla del mapa puede corresponder a uno de cuatro tipos de terreno:

- **Camino:** accesible tanto para el jugador como para los cazadores.
- **Lianas:** zonas por las que solo los cazadores pueden desplazarse; representan obstáculos para el jugador.
- **Túneles:** áreas exclusivas para el jugador, inaccesibles para los cazadores.
- **Muros:** bloquean el paso tanto del jugador como de los cazadores.

Modo 1: Escapa

En este modo, el jugador debe huir de un grupo de cazadores a través de un laberinto. Si alguno de los cazadores logra alcanzarlo, el jugador pierde la partida. En cambio, si consigue llegar a la salida antes de ser atrapado, gana.

El puntaje del jugador dependerá del tiempo que tarda en escapar: a menor tiempo, mayor puntuación. Además, el nivel de dificultad influirá directamente en los puntos obtenidos, ya que un mayor número de cazadores y una mayor velocidad de estos incrementarán la recompensa. El esquema de puntaje debe ser definido por el grupo e indicado en la documentación.

Modo 2: Cazador

En este segundo modo, las reglas se invierten: el jugador se convierte en el cazador y su objetivo es atrapar a los demás cazadores que aparecerán en el mapa. Cada vez que un cazador es atrapado, otro debe generarse en una nueva posición.

La dificultad también puede ajustarse aumentando la velocidad de los enemigos. Si un cazador logra alcanzar una de las salidas, el jugador perderá puntos (la cantidad será determinada por el grupo). Por el contrario, si el jugador logra atraparlo antes de que escape, obtendrá el doble de los puntos que perdería si el enemigo saliera.

Requisitos funcionales

- ### Movimiento del jugador:

  El jugador debe poder desplazarse en las cuatro direcciones (arriba, abajo, izquierda y derecha). Además, debe contar con la habilidad de correr, aunque esta acción estará limitada por una barra de energía.

  La gestión de la energía (su consumo, recuperación y visualización) queda a criterio del grupo, pero debe existir una forma visible en pantalla que permite al jugador conocer su nivel de energía actual en todo momento.

- ### Trampas (modo Escapa):

  En el modo Escapa, el jugador tendrá la opción de colocar trampas en el mapa. Estas trampas funcionan como una herramienta estratégica para eliminar enemigos.

  Las condiciones de uso son las siguientes:

  - El jugador puede tener un máximo de tres trampas activas en el mapa simultáneamente.
  - Cada trampa tiene un tiempo de reutilización de 5 segundos antes de poder colocarse nuevamente.
  - Si un cazador pasa por una trampa, muere inmediatamente y la trampa desaparece del mapa.
  - Por cada cazador eliminado, el jugador recibe un pequeño bono de puntos adicional.
  - Los cazadores eliminados reaparecen luego de 10 segundos, manteniendo el equilibrio de dificultad.

Esta mecánica debe estar correctamente sincronizada con el sistema de puntuación y el control del mapa, de modo que no interfiera con la fluidez del juego ni genere errores de colisión o reaparición.

- ## Mapa del juego:

El mapa deberá estar representado mediante una **matriz numérica**, donde cada celda contiene un valor que identifica el tipo de terreno. El comportamiento tanto del jugador como de los enemigos dependerá del tipo de casilla en la que se encuentren.

Para garantizar una estructura ordenada y modular, **cada tipo de casilla** (camino, muro, túnel o liana) deberá implementarse como una **clase independiente**, con su propio comportamiento y sus respectivas restricciones de movimiento.

Además, el mapa debe **generarse de manera aleatoria** cada vez que se inicie una nueva partida. Es decir, **cada partida debe contar con un mapa distinto**, construido dinámicamente.

La generación del mapa debe **asegurar siempre la existencia de al menos un camino válido** que permita al jugador llegar desde su posición inicial hasta la salida.

- ## Enemigos:

Se debe implementar una clase denominada Enemigo, la cual controlará el comportamiento de los cazadores.

- En el modo **Escapa**, los enemigos deben perseguir al jugador.
- En el modo **Cazador,** los enemigos deben huir del jugador.

En ambos casos, los enemigos deben poder desplazarse en las cuatro direcciones al igual que el jugador, respetando las mismas restricciones de movimiento según el tipo de casilla.

- **Registro de jugadores:**

  Antes de iniciar la partida, el juego debe mostrar una ventana de registro donde el jugador escriba su nombre.

  Este nombre se utilizará para:

  - asociar la puntuación del jugador,
  - mostrarlo en las listas de puntajes,
  - y mantener un historial básico de jugadores.

  El registro debe ser obligatorio antes de comenzar cualquiera de los dos modos.

- **Puntajes:**

  El juego debe incluir un sistema de puntuaciones que muestre dos listas separadas:

  - Top 5 – Modo Escapa
  - Top 5 – Modo Cazador

  Cada lista debe:

  - mostrar el nombre del jugador,
  - su puntaje,
  - y estar ordenada de mayor a menor.

  El sistema debe actualizarse automáticamente cuando un nuevo puntaje entre dentro de los primeros cinco lugares.

Repositorio en GitHub

El proyecto debe mantenerse en un repositorio de GitHub, el cual debe mostrar evidencia de un trabajo progresivo y continuo mediante el registro de commits.

Se evaluará que:

- Los commits sean frecuentes y no únicamente al final del proyecto.

- Los mensajes de commit sean claros y describan adecuadamente los cambios realizados.

- El historial refleje la participación del grupo y la evolución del desarrollo del sistema.

El repositorio debe mantenerse público o compartido con el docente para efectos de revisión.

Documentación del Proyecto

Cada grupo deberá realizar **un único documento** con una extensión máxima de dos páginas, que contenga las siguientes dos secciones.

Debe aparecer el inciso y la respuesta correspondiente en cada caso.

1. Atributo de Análisis de Problema:

Indique de manera clara cuál es el problema complejo de ingeniería, utilizando principios de matemáticas, ciencias naturales y ciencias de la ingeniería, integrando además aspectos relacionados con el desarrollo sostenible.

Realice un análisis del contexto y de las variables relacionadas con el problema complejo de ingeniería identificado, integrando los aspectos correspondientes al desarrollo sostenible.

Describa un plan de solución para el problema complejo de ingeniería, integrando aspectos para el desarrollo sostenible. Incluya la manera en que se resuelve el problema, considerando los principios de ingeniería y sostenibilidad.

Evalúe, con pros y contras, las soluciones planteadas al problema complejo de ingeniería, integrando aspectos relacionados con el desarrollo sostenible.

2. Atributo de Herramientas de Ingeniería:

Explique las técnicas, recursos, herramientas o métodos acordes con las variables del problema complejo de ingeniería que se utilizan para resolver el problema. Debe realizar un diagrama de clases del modelo de objetos utilizado.

Indique cómo aplicar las técnicas, recursos, herramientas o métodos para la realización del proyecto.

Explique cómo adapta las técnicas, recursos, herramientas o métodos durante el desarrollo del proyecto.

Detalles de la Entrega

Repositorio de GitHub (OBLIGATORIO)

El repositorio debe incluir:

- Todo el código del proyecto, funcional y organizado.
- La documentación del proyecto (archivo PDF o Word dentro del repositorio).
- Evidencia de trabajo progresivo mediante commits frecuentes y con mensajes claros.
- Estructura ordenada de carpetas.
- Debe ser público.
- Si el repositorio no se entrega o no se envía mediante el tec digital, no se evalúa el proyecto.

Evaluación del Proyecto – Juego “Escapa / Cazador”

Funcionalidad de la aplicación – 75%

Modo Escapa – 15%

Modo Cazador – 15%

Mapa y clases de terreno – 15%

Movimiento, energía y física básica – 10%

Interfaz, sonido y animaciones – 10%

Dificultad y balance – 10%

Documentación externa e interna – 25%

Documentación técnica – 10%

Diagrama de clases – 5%

Comentarios y orden – 5%

Presentación – 5%