

# TRON

Instituto Tecnológico de Costa Rica  
Escuela de Ingeniería en Computadores  
Algoritmos y Estructuras de Datos II (CE 1103)  
II Semestre 2024

## OBJETIVOS

### GENERAL

- Implementar una solución a un problema mediante la utilización de estructuras de datos lineales.

### ESPECÍFICOS

- Implementar las diferentes estructuras de datos lineales (listas, pilas y colas).
- Desarrollar algoritmos para dar solución a un problema.
- Implementar un mecanismo para manipular las estructuras de datos implementadas.
- Fomentar la creatividad mediante el análisis y diseño de algoritmos
- Utilizar diagramas de clases UML para modelar una solución a un problema.
- Aplicar patrones de diseño en la elaboración de una solución a un problema.

## REQUERIMIENTOS

Tron es un juego de carreras de motos de luz. El mapa consiste en una malla en la que pueden navegar las motos de luz en 4 direcciones posibles. Cada jugador maneja una moto de luz (utilizando las flechas del teclado) en la malla. Las motos dejan una estela destructiva a su paso. Si otra moto de luz cruza una estela dejada por otra moto de luz, esta se destruye.

ID	DESCRIPCIÓN	PUNTOS
001	Las motos de luz se implementan como una lista enlazada simple. Cada moto deja una estela destructiva a su paso. El movimiento de las motos se puede asemejar al de una oruga. Cuando la moto se crea, inicialmente tendrá una estela de 3 posiciones.	15
002	Las motos tienen los siguientes atributos: <ul style="list-style-type: none"><li>• <b>Velocidad:</b> valor aleatorio entre 1 y 10 que determina qué tan rápido una moto se mueve</li><li>• <b>Tamaño de la estela:</b> valor que determina el largo de la estela. Inicialmente vale 3.</li></ul>	10

	<ul style="list-style-type: none"> <li>• <b>Combustible:</b> valor que determina cuánto combustible tiene la moto. Se consume automáticamente dependiendo de la velocidad de la moto a una tasa de 1 celda de combustible por cada 5 elementos de la malla recorridos. Es un valor de 0 a 100.</li> <li>• <b>Ítems:</b> cola de elementos que afectan permanentemente la moto</li> <li>• <b>Poderes:</b> pila de poderes que afectan temporalmente la moto</li> </ul>	
003	Cuando una moto se destruye los ítems y poderes que tenía aun sin usar, se colocan en el mapa en posiciones aleatorias.	5
004	El jugador escoge cuándo ejecutar los poderes, los cuales se ejecutan en un orden definido por el jugador. En pantalla, el jugador podrá ver la pila de poderes. Presionando un botón puede ir moviendo el elemento del tope de la pila para dejar el poder que más le convenga de primero. Cuando presione el botón de aplicar el poder, se aplicará siempre el elemento del tope.	10
005	Los ítems se aplican en el orden de llegada automáticamente con un <i>delay</i> de 1 segundo entre la aplicación de uno y otro, aplicando prioritariamente las celdas de combustible. Si el combustible está lleno, la celda se vuelve a insertar en la cola sin aplicarse.	5
006	Una moto se destruye al chocar con otro jugador (ambos mueren), cruzar una estela o quedarse sin combustible. Las motos nunca se detienen. El jugador únicamente puede cambiarlas de dirección.	10
007	Tal y como se indicó el mapa es un <i>grid</i> o malla de tamaño fijo. Se implementará mediante una lista enlazada en la que cada nodo posee 4 referencias a otros nodos, formando así la red. Cuando el juego inicia, se carga el mapa de un tamaño previamente definido. El jugador utiliza las flechas del teclado para mover la moto en el <i>grid</i> .	15
008	<p>En la red aparece ítems y poderes aleatoriamente, que pueden recoger el jugador. Los ítems incluyen:</p> <ul style="list-style-type: none"> <li>• <b>Celda de combustible:</b> incrementa el combustible de la moto. Cada celda tiene una capacidad aleatoria.</li> <li>• <b>Crecimiento de estela:</b> incrementa el tamaño de la estela en un tamaño variable. Cada ítem tiene un valor aleatorio de 1 a 10 que determina cuánto va a incrementar la estela.</li> <li>• <b>Bombas:</b> cuando un jugador toma una bomba, explota.</li> </ul> <p>Los poderes incluyen:</p> <ul style="list-style-type: none"> <li>• <b>Escudo:</b> permite que la moto se haga invencible por un tiempo variable. Afecta visualmente la moto.</li> <li>• <b>Hiper velocidad:</b> aumenta la velocidad de la moto en un valor aleatorio y por un periodo aleatorio. Afecta visualmente la moto.</li> </ul>	10
009	Existen <i>bots</i> que simulan otros jugadores. Todas las reglas anteriores aplican para los <i>bots</i> . Su comportamiento es aleatorio. Al menos 4 bots simultáneos deberán aparecer en el juego	20
010	El juego se programará en C# con interfaz gráfica en Windows Forms/MAUI/Unity	BINARIO
011	El estudiante debe implementar todas las estructuras de datos requeridas en el proyecto.	BINARIO
012	Se evaluarán buenas prácticas de programación en el código, las cuales es responsabilidad del estudiante investigar y aplicar, esto incluye patrones de diseño	A CRITERIO DEL PROFESOR

## ASPECTOS OPERATIVOS

- El trabajo se realizará de forma **individual**
- El uso de Git y Github es obligatorio
- La fecha de entrega será según lo especificado en el TEC Digital. Se entrega en el TEC digital, un archivo PDF con la documentación. Los estudiantes pueden seguir trabajando en el código hasta 15 minutos antes de la cita revisión oficial.

## DOCUMENTACIÓN

- La documentación deberá tener las partes estándar:
  - Portada
  - Introducción
  - Tabla de contenidos (con los títulos debidamente numerados)
  - Breve descripción del problema
  - Descripción de la solución
    - Por cada uno de los requerimientos, se deberá explicar cómo se implementó, alternativas consideradas, limitaciones, problemas encontrados y cualquier otro aspecto relevante.
  - Diseño general: diagrama de clases UML con las clases relevantes que muestren el diseño orientado a objetos y los patrones de diseño aplicados

## EVALUACIÓN

- El proyecto tiene un valor de 25% de la nota del curso
- Los proyectos que no cumplan con los siguientes requisitos no serán revisados:
  - Toda la solución debe estar integrada
  - La interfaz de usuario debe estar implementada e integrada
- El código tendrá un valor total de 80%, la documentación 10% y la defensa 10%. De estas notas se calculará la *Nota Final del Proyecto*.
- Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
  - **Si no se entrega documentación en formato PDF, automáticamente se obtiene una nota de 0.**
  - Si no se utiliza un manejador de código se obtiene una nota de 0.
  - Si la documentación no se entrega en la fecha indicada se obtiene una nota de 0.
  - El código debe desarrollarse en C#, si no, se obtendrá una nota de 0.
- La revisión de la documentación será realizada por parte del profesor, no durante la defensa del proyecto.
- Cada estudiante tendrá 20 minutos para exponer su trabajo al profesor y defenderlo, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo que se recomienda tener todo listo antes de entrar a la defensa.

- Cada grupo es responsable de llevar los equipos requeridos para la revisión, si no cuentan con estos deberán avisar al menos 2 días antes de la revisión a el profesor para coordinar el préstamo de estos.
- Durante la revisión únicamente podrán participar los miembros del grupo, asistentes, otros profesores y el coordinador del área.