



Trabajo práctico

Objetivo: Completar la lógica de un juego estilo *Tower Defense* mediante la implementación y uso de estructuras de datos dinámicas, y diseñar una estrategia que maximice la eficiencia en la colocación de torres para eliminar enemigos.

Material Provisto

- **Descripción del proyecto:** Se detalla cómo funciona el juego y los objetivos generales del mismo.
- **Simulador de niveles:** Permite observar cómo se desarrolla una partida. Lee los datos de un nivel y una estrategia, ejecuta el juego y muestra los resultados visualmente.
- **Estrategia inicial básica:** Algoritmo sencillo que ubica torres aleatoriamente.

Tareas a Realizar

- Diseñar una estrategia con backtracking.**
Diseñar una estrategia de colocación de torres con la técnica de *backtracking*. La cual consiste en tomar una serie de decisiones y "arrepentirse" o (es decir, deshacer la última) en caso de que la decisión no contribuya al objetivo del problema.
- Diseñar una estrategia con backtracking.**
Diseñar una estrategia de colocación de torres de su autoría. Puede considerar distintos factores como la cantidad y vida de enemigos, la cobertura del mapa, etc.
- Proveer implementaciones de las estructuras de datos auxiliares**
Se alienta a implementar y testear las estructuras de datos que se utilicen de manera independiente al juego.
- Integrar y probar.**
Asegurarse de que la estrategia funcione correctamente con el simulador. Verificar que los enemigos sean eliminados de manera eficiente y que las torres se comporten de forma coherente.
- Redactar un informe breve.**
Explicar el funcionamiento de las estructuras de datos implementadas. Justificar las decisiones tomadas en el diseño de la estrategia. Incluir dificultades encontradas y posibles mejoras.

Descripción del Juego

El juego consiste en defender un camino por el cual avanzan enemigos, colocando torres que los atacan a medida que pasan por su rango. Los niveles que se proveen presentan escenarios de dificultades variadas. Se debe diseñar una estrategia previa a la ejecución del nivel que determine en qué posiciones del mapa se colocarán las torres. Una vez iniciada la simulación, no se puede cambiar la estrategia.

Cada **nivel** está definido por un archivo de entrada ubicado en la carpeta *Levels*. Se deben probar las estrategias sobre todos los niveles provistos e inclusive se puede entregar una serie de su autoría.

Componentes del nivel

Cada nivel contiene:

- Un **mapa** rectangular compuesto por casillas:
 - Casillas de camino, por donde se desplazan los enemigos.
 - Casillas disponibles para colocar torres.
 - Casillas bloqueadas que solo sirven de relleno.
- Una **cantidad de enemigos** que determina el número de enemigos de la horda.
- Una **cantidad de torres** que determina cuántas torres se pueden colocar por nivel.
- Una **distancia de ataque** que representa el radio del área en la que ataca cada torre.

Mecánica del juego

- Los enemigos entran por la casilla superior izquierda del mapa y siguen el camino hasta la salida. Se desplazan una unidad en el mapa por cada unidad de tiempo.
- Las torres disparan automáticamente cuando un enemigo entra en su rango de ataque y lo hacen en cada unidad de tiempo. El daño se aplica de forma inmediata.
- Si un enemigo pierde toda su vida, se elimina del nivel. Si logra llegar a la salida, el juego termina. Si todos los enemigos se eliminan también.

Simulación

El simulador ejecuta el nivel utilizando la estrategia definida por el estudiante y muestra una representación de la partida, es decir, la posición de las torres y la posición y vida de los enemigos.

Entregables

- Archivos `.c` y `.h` con las estructuras implementadas y la estrategia.
- `informe.pdf` (máximo 1 página).
- Opcional: niveles de prueba adicionales que muestren casos en donde la estrategia propuesta mejora.

Criterios de Evaluación

Criterio	Puntos
Implementación de estructuras	20
Integración con la estrategia	10
Funcionamiento de la estrategia	30
Estilo y claridad del código	20
Informe y justificación técnica	10
Defensa	10
Total	100