

Taller de Programación, informe.

Manuel Slepoy, Mateo Delmagro.

Observaciones.

1. Todos los enemigos tienen la misma cantidad de puntos de vida.
2. Todas las torres realizan 1 punto de daño cada turno, dentro de una área constante y definida.

Teoremas.

- A partir de las observaciones (1) y (2) tenemos que: derrotar a un solo enemigo es lo mismo que derrotarlos a todos.
- A partir de la observación (2): una torre le hará a un enemigo tanto daño como la cantidad de celdas de camino dentro de su área de ataque.
- Partiendo del teorema anterior, para cualquier casilla donde podría haber una torre, podemos precomputar el impacto potencial que la tendría sin colocarla.

Estrategia con backtracking.

- Para poder arrepentirse de un paso, organizamos una pila con “tramas” que representan el estado en el que estábamos antes de tomar o no una decisión.
- Si en cualquier paso no existe forma de superar el daño que ya sabemos que otra configuración logró realizar, abandonamos ese camino. Detectamos esto heurísticamente con una tabla aditiva de los impactos potenciales en adelante.

Estrategia custom.

- Tomamos la lista de impactos potenciales descrita en el Teorema 3, y las ordenamos de mayor a menor.
- Recorriendo esa lista ordenada, tomamos tantas torres como sea posible, maximizando el daño que realizamos.

Dificultades y mejoras.

- La codebase entregada tomaba por convención al eje X como eje vertical, lo que consideramos extraño. (en serio profe WHYYYY)
- Muchas de las construcciones que utilizamos para la implementación de nuestras estrategias necesitaban datos que simplemente no eran triviales de acceder con la estructura que nos fue entregada—por ejemplo, la cantidad de casillas vacías en el mapa. Esto no es el fin del mundo, pero tener que computarlo toma su tiempo.