

Función eval()

Como funciona

Nuestra función de evaluación utiliza varios factores (cada uno con su propio peso/importancia) para determinar la puntuación de un estado:

- Cantidad de piezas vivas:
 - Recorremos el tablero y contamos la cantidad de piezas vivas del agente y del rival, sumando al score la resta de ambas (piezas_propias – piezas_rival).
- Posición de los masters propio y rival:
 - En caso de no encontrar alguno de los dos, devuelve INF o -INF para determinar si ganó o si perdió el agente ($INF = 10^6$).
 - En caso de que alguno de los masters llegue al shrine, de la misma forma, devuelve INF o -INF para mostrar si el master del agente llegó o si el master rival llegó.
- Control del centro:
 - Le damos una pequeña pero útil importancia a esta variable, contando la cantidad de peones (no master) que se encuentran en las posiciones centrales (B3, C3 y D3).

Justificación

Elegimos estos tres factores ya que al ser un tablero pequeño y tener un tiempo limitado de 50ms por jugada, nos pareció la opción más competitiva dentro de los límites de tiempo.

- Contando la cantidad de piezas vivas, y teniendo en cuenta que en el onitama existen solo 5 piezas las cuales pueden moverse de igual manera todas dependiendo de la carta, nos pareció lo mejor mantener una mayor cantidad de piezas que el rival en todo momento posible.
- Con la posición de los masters podemos fácilmente decidir si la partida termino en victoria o en derrota sobre ese estado.
- Utilizando un control del centro podemos asegurar mayor cantidad de acciones posibles por jugada por libertad de movimiento y protección de las demás piezas.

Aclaración

Utilizamos una segunda función de eval() llamada eval_simplificada() para ordenar las acciones cuando se generan de forma rápida contando la cantidad de piezas vivas del agente menos cantidad de piezas vivas del rival. Esto permite realizar la poda alpha-beta antes y se obtiene un mejor rendimiento general.