

Jugador de Hex

Lia Stephanie López Rosales

April 11, 2025

1 Introducción

El jugador implementado utiliza una combinación de:

- Heurísticas posicionales
- Búsqueda Minimax con poda alfa-beta
- Ajustes dinámicos tanto en las heurísticas como en la búsqueda

2 Heurísticas

2.1 Peso a los Bordes (edge_weights)

Objetivo: Favorece celdas cercanas a los bordes objetivo.

Implementación:

$$\text{edge_weights}(i, j) = \begin{cases} \min(y + 1, \text{size} - y) & \text{jugador horizontal} \\ \min(x + 1, \text{size} - x) & \text{jugador vertical} \end{cases}$$

Efectividad: Controlar celdas adyacentes a los bordes propios permite construir puentes ganadores posteriormente.

2.2 Peso al Centro (center_weights)

Objetivo: Valora posiciones centrales estratégicas.

Implementación:

$$\text{center_weights}(i, j) = \frac{1}{1 + \sqrt{(i - c)^2 + (j - c)^2}}, \quad c = \frac{\text{size} - 1}{2}$$

Efectividad: Una celda central tiene 6 vecinos vs 3 en bordes, maximizando opciones de expansión.

2.3 Bonificación por Puentes (bridge_bonus)

Objetivo: Incentiva conexiones diagonales que forman "puentes".

Implementación:

$$\text{bridge_bonus}(i, j) = \begin{cases} 2 & (i + j) \equiv 0 \pmod{2} \\ 1 & \text{otherwise} \end{cases}$$

Efectividad: Dos piezas en puente crean caminos duales \Rightarrow inmunidad a bloqueos simples.

2.4 Penalización por Cercanía al Oponente (opponent_penalty)

Objetivo: Desincentivar la colocación de piezas en zonas controladas por el oponente.

Implementación:

$$\text{opponent_penalty}(i, j) = 1.5 - 0.5 \cdot \frac{|i - j|}{\text{size}}$$

Efectividad: Evitar intentar conectar bpor una zona fácilmente bloqueable o donde ya no es posible conectar

3 Función de Evaluación

Combina componentes con pesos adaptativos:

$$\text{Valor} = 0.6P + 0.3C + 0.1A + 0.3M$$

donde:

P = Control posicional (Bordes)

C = Conexiones potenciales (Bonificación por Puentes)

A = Control de área (Penalización por Cercanía al Oponente)

M = Influencia central

4 Ordenamiento de Movimientos

Algoritmo de priorización:

1. Calcular score por celda:

$$\text{score} = \underbrace{w_e}_{\text{borde}} + \underbrace{w_c}_{\text{centro}} + \underbrace{0.8w_b}_{\text{puente}} - \underbrace{0.6w_o}_{\text{oponente}}$$

2. Ordenar descendente
3. Podar el 40% inferior en fase tardía

5 Profundidad Iterativa

Para simultáneamente no incumplir con las restricciones de tiempo y aprovecharlo todo lo posible, se implementa un sistema que evalúa antes de aumentar la profundidad de búsqueda el tiempo disponible, devolviendo la mejor jugada alcanzada según el tiempo dado en vez de tener una profundidad prefijada.