Hex Ia

Dario Lopez Falcon

12 de abril de 2025

1. Estructura del poryecto

1.1. Clases:

El plroyecto consta solo de dos clases:
`1-class MinMax_Player(Player):`
Esta es la clase principal que cuenta con los metodos necesarios para generar las jugadas segun una configuracion de tablero y un usuario.

Para crear una instancia de la misma se debe llamar al metodo `MinMax_Player(id)`, donde id es 1 o 2 (identificador del jugador).

Esta clase cuenta con el metodo `play`, que recibe como parametros a `board`: HexBoard, y `time`: int. Este metodo se encarga de generar una jugada segun una configuracion de tablero, usando el algoritmo de `minimax` y aplpicando la `poda alpha-beta` para optimizar

`2- class DSU(Player):`

Esta clase es una implementacion de la estructura de datos DSU usada para manejar de manera eficiente conjuntos disjuntos. Es usada en el calculo de una de las heuristicas.

Para facilitar su `uso` se concentro toda la logica en un solo archivo (`player.py`)

2. ¿Como se usa?

Toda la logica se encuentra concentrada en la clase `MinMax_Player` dentro de `player.py`, por lo que primero debemos importar la clase de la libreria usando `from player.py import MinMax_Player`

Luego solo faltaria crear una instancia de dicha clase llamando al metodo `MinMax_Player`

Ejemplo:

$$player = MinMax\_Player(1)$$

Para generar jugadas usamos el metodo `play` pasandole un tablero y un limited de tiempo, el cual es opcional y por defecto es 10s

Ejemplo:

$$player.play(board, 2)$$

esto devolvera una tupla: (fila, columna) la cual hace referencia a la jugada generada por la ia.

3. ¿Cómo funciona?

Como el juego de Hex es un problema se suma cero, es ideal atacarlo usando el algoritmo de Minimax con poda apha-beta.
`Minimax:`

1-Es un algoritmo para juegos de dos jugadores (ej: Hex) que maximiza las ganancias del jugador actual y minimiza las del oponente.

2-Explora un árbol de posibles movimientos alternando entre capas MAX (elige el mejor valor) y MIN (elige el peor valor para MAX).
`Poda Alpha-Beta:`

- Optimización de Minimax que reduce el número de nodos evaluados.
- Alpha α: Mejor valor encontrado para MAX *inicial* : $-\infty$.
- Beta β: Mejor valor encontrado para MIN *inicial* : $+\infty$.
- Podar (cortar) ramas cuando:
  - En capa MAX: si un valor $\geq \beta$ (el rival MIN no permitirá esta rama).
  - En capa MIN: si un valor $\leq \alpha$ (MAX ya tiene una opción mejor).

Resumen:
Minimax busca la mejor jugada evaluando todas las posibilidades, mientras que Alpha-Beta acelera el proceso descartando ramas inútiles sin evaluarlas.

4. Heurísticas Principales

El código implementa varias heurísticas para evaluar movimientos en el juego Hex, cada una con un peso específico:

- **Puentes (bridges)**: Detecta patrones que forman conexiones estratégicas (peso: 100)
- **Cierre de puentes (close_bridge)**: Cierra puentes del jugador cuando podrian ser bloqueados por el oponente (peso: 200)
- **Centro (center)**: Favorece movimientos hacia el centro del tablero (peso: 5)
- **Camino completo (complete_road)**: Completa conexiones ganadoras (peso: 100)
- **Victoria (win)**: Movimiento ganador inmediato (peso: 100000)

- **Bloqueo (block_enemy)**: Interrumpe jugadas estratégicas del oponente (Cerrar puentes del oponente) (peso: 100)
- **Vecinos (neighbor)**: Penaliza movimientos con muchos vecinos enemigos (peso: -1000)
- **Estructura de unión (dsu)**: Evalúa conexiones potenciales usando estructura de datos union-find (peso: 2)

5. Funciones Clave

- `h()`: Función principal que combina todas las heurísticas
- `bridges()`: Detecta patrones de puente predefinidos
- `close_bridge()`: Identifica oportunidades para cerrar puentes enemigos
- `calculate()`: Implementa algoritmo DSU para evaluar conexiones
- `bfs()`: Usa búsqueda en amplitud para evaluar caminos potenciales

6. Conclusión

El sistema utiliza múltiples heurísticas con diferentes pesos para evaluar movimientos, combinando estrategias ofensivas (creación de caminos) y defensivas (bloqueo). La función principal `h()` agrega estos componentes para producir una evaluación global de cada movimiento potencial.