

Informe Escrito

Adrián Hernández Castellanos C312

HexAIPlayer: Jugador virtual para el juego de mesa HEX

El presente informe contiene toda la información importante a tener en cuenta respecto a la clase de jugador implementada, donde el objetivo es elegir la mejor jugada disponible dado un estado del tablero usando IA y heurísticas.

ESTRUCTURA DEL PROYECTO:

El proyecto cuenta con 4 archivos fundamentales:

- Los archivos `base_class_player.py` y `base_class_hexboard.py` contienen las implementaciones de las clases base mencionadas en la orientación del proyecto. Si se desea sustituirlas por otras de implementación propia solo se deben cambiar las líneas de `import` en `player.py`
- El archivo `game.py` contiene una ejecución del juego Hex en consola, donde se inicializan los jugadores virtuales y se realiza el enfrentamiento. Similar al ejemplo anterior, este archivo es reemplazable y no necesita realizarse ninguna modificación en `game.py`
- El archivo `player.py` contiene la clase `HexAIPlayer`, que hereda de la clase base `Player` e implementa los métodos de constructor y `play()` el cual ejecuta la jugada deseada. Al constructor se le debe pasar como parámetros un entero que representa el ID del jugador que asumirá la IA (1->Lados izquierdo y derecho) (2->Lados arriba y abajo) y un entero que representa el tiempo límite que puede demorar la jugada en realizarse.

ESTRATEGIA Y MÉTODOS DE LA CLASE:

La estrategia seguida es una combinación de varias heurísticas que permiten evaluar las jugadas que puede realizar el jugador y elige la mejor estimada. Para esto se usa la combinación de estrategias Minimax con podas alpha-beta y A* con heurística Manhattan Distance adaptada, además de otras técnicas de ponderación de jugadas que permiten comparar estados del juego en base a una puntuación estimada. La estrategia sigue el siguiente flujo:

- Inicialmente se ejecuta minimax, donde se evalúa cada jugada disponible por el jugador virtual, y por cada jugada, se analiza cada jugada posible del oponente, tras lo cual se repite el ciclo hasta llegar a una profundidad especificada. Al llegar a dicha profundidad, se devuelve un valor que representa a la jugada. Este valor es una puntuación del estado general del tablero hasta ese punto, y posteriormente se comparan estas puntuaciones, tanto las del jugador (alpha) como las del oponente (beta) y se realizan podas en estados intermedios del análisis para evitar analizar jugadas no necesarias.
- Para obtener la puntuación de cada jugada, minimax llama a la función `evaluate`, que ejecuta varias heurísticas que otorgan o reducen el peso de la puntuación del tablero actual para el jugador. A continuación se explicará cada una de estas heurísticas:
 - `A_Star` con `Manhattan_Distance`: Calcula el costo de completar el camino de costo mínimo y lo devuelve. Este costo en la función de evaluación es convertido posteriormente en un score que es inversamente proporcional a dicho costo (mientras menor costo, mayor score)

- Score básico por cantidad: Calcula celdas del jugador y celdas del oponente, sumando y restando la misma cantidad de puntos por cada una. Esto define una ventaja muy básica para el jugador que empezó el juego.
 - Centralidad y vecindad: Cada ficha vacía que esté en el centro aumenta la puntuación, pues las fichas vacías pueden ser llenadas por el jugador como parte de la estrategia de propagación multidireccional. Esto no ocurre así en las paredes o esquinas, ya que poseen menos direcciones hacia las cuales expandirse. Además, cada una de esas fichas recibe bonificación si sus alrededores pertenecen al jugador y se penalizan si sus alrededores pertenecen al oponente.
 - Bloqueo estratégico: Se suman puntos por cada ficha que bloquee una posible posición de puente entre dos casillas del oponente si estas van en la dirección que debe ir el oponente.
 - Evaluación de cadenas: Otorga bonificación por alargar una cadena propia ya existente y penaliza si el oponente alarga una cadenasuya.
- Además, existe una función que ejecuta una táctica de fallback cuando no se detectan caminos posibles para ganar, lo cual valorala jugada que más perjudica al oponente y la devuelve.

CARACTERÍSTICAS EXTRA:

- Cada función mencionada anteriormente tiene una distribución de puntuación diferente en dependencia de con qué enfoque se quiere que la IA juegue (a la defensiva, priorizar ganar, etc).
- La profundidad con la que analiza minimax se obtiene de manera dinámica en dependencia del estado actual del tablero.
- Los movimientos que minimax analiza son previamente ordenados usando las heurísticas de ponderación de jugadas en el primer nivel. Esto ocurre con el objetivo de aumentar las probabilidades de encontrar más rápido a la mejor jugada.