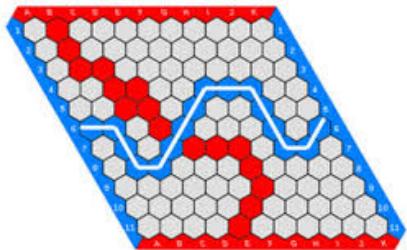


Agente minimax para Hex

- Samuel David Suárez Rodríguez C-412
 - Enmanuel Verdesia Suárez C-411
-

Hex es un juego entre dos jugadores que toman turnos colocando fichas en un tablero romboidal, compuesto por casillas hexagonales. Las fichas de cada jugador se distinguen por su color, y gana el jugador que consiga conectar dos laterales opuestos del tablero previamente asignados.

Reglas del juego:



Ejemplo de estado del juego
(Ganan las azules)

- **Inicialmente el tablero está vacío. A cada jugador se le asigna un color de fichas y dos laterales opuestos del tablero que tendrá que intentar conectar con sus fichas siguiendo las reglas del juego.**
- **Los jugadores van colocando fichas por turnos sobre el tablero en casillas desocupadas.**
- **Gana el primer jugador que consigue formar una línea de sus fichas que**

conecte sus dos laterales. No son posibles los empates.

- **Como el primer jugador tiene ventaja, se jugarán 2 partidas una con cada color de fichas.**

Descripción del agente

El agente desarrollado implementa minimax con una profundidad limitada a partir de una función heurística debido a que el branch factor del árbol de decisión del juego es muy grande sobre todo en fases iniciales de la partida.

Estados y transiciones

Nodo

Según la representación del juego en el árbol de decisión, un nodo en este árbol corresponde a un estado del juego, que está determinado por las fichas jugadas por ambos jugadores y de donde se puede inferir el turno del jugador.

Arista

Las aristas representan transiciones entre estados, o sea, jugadas de algún jugador, las cuales corresponden a poner una ficha en el tablero, de esta manera una arista conecta dos nodos, el nodo antes de jugar la ficha (padre) y el nodo posterior a jugar la ficha (hijo).

Camino

Un camino está formado por una secuencia de aristas pertenecientes a los nodos correspondientes, el cual se puede ver como una simulación del juego desde algún estado inicial (comienzo del camino) y hasta algún estado (fin del camino) no necesariamente final.

Jugada

Una jugada es una decisión tomada a partir de determinado estado, o sea, la elección de una arista a partir del conjunto de aristas de un nodo en el árbol de decisión.

Jugador

El jugador es el ente abstracto responsable de la toma de decisiones en determinada jugada, en este juego un jugador es responsable de las jugadas que se realizan desde nodos en profundidad par, y el otro jugador desde los impares.

Heurística

La función heurística es la encargada de proveer al usuario información sobre qué tan bueno o malo es un estado determinado, lo cual puede ayudar a la toma de decisiones en determinados momentos del juego en los que no conviene recorrer todo el espacio de búsqueda. Estas funciones a menudo son inexactas, pues solo aproximan la calidad de los estados a partir de la experiencia humana generalmente, sin embargo, si se combina con otras técnicas como minimax (con profundidad limitada), la cual es capaz de observar varias decisiones, puede tener un mejora considerable en el performance del agente.

Para el desarrollo de este agente, se hizo una observación principal del juego:

- **Si la mínima cantidad de fichas que necesita poner el jugador actual para ganar es menor que la del adversario, el jugador actual se encuentra en una buena posición.**

Aunque esto no tiene sentido en algunos casos, sí lo tiene en la mayoría, por lo que se puede usar como una aproximación del estado del juego, y sobre todo ahorra tiempo de exploración en el árbol de decisión. Para calcular la cantidad mínima de fichas necesarias para ganar se utilizó una **Búsqueda de Costo Uniforme (BCU)**, específicamente el caso de algoritmo de **Dijkstra**.

Búsqueda de la cantidad mínima de fichas

El algoritmo de **Dijkstra** es una variante de la **Búsqueda de Costo Uniforme** cuando los pesos de las aristas en el grafo son no negativos, de esta manera podemos asumir que en cada momento el camino mínimo parcial se ha encontrado una vez que se saca el nodo de la cola de prioridad. Para nuestro problema queremos saber la cantidad mínima de fichas necesarias para conectar dos lados opuestos teniendo en cuenta las fichas que ya hay jugadas. Para esto basta con transformar el tablero del juego en un grafo donde las celdas son los nodos y las aristas conectan nodos adyacentes (sin incluir nodos con fichas opuestas), una arista tiene costo 0 si se dirige a un nodo destino que representa una ficha del jugador actual, y costo 1 en caso contrario. Si creamos un nodo artificial que conecte todos los nodos desde un lado del tablero y calculamos el camino mínimo hacia algún nodo del otro lado tenemos la respuesta al problema.