

Jugador inteligente para el juego de mesa Hex

Autor

Victor Hugo Pacheco Fonseca

11 de abril de 2025

Índice

1. Introducción	2
2. Hex	2
2.1. Objetivo del juego	2
2.2. Reglas	2
3. Algoritmo Implementado	2
4. Heurísticas	2
4.1. Dijkstra	2
4.2. Incorporación de puentes	3
4.3. Evaluación de Puentes y Posiciones Críticas	3
4.3.1. Detección de puentes	3
4.3.2. Verificación de posiciones críticas	3

1 Introducción

El objetivo de este proyecto es desarrollar una IA para el juego de Hex, que tenga la capacidad de evaluar posiciones complejas y tomar decisiones informadas durante la partida. Para ello se han implementado diversas estrategias de búsqueda y evaluación, combinando algoritmos clásicos de inteligencia artificial y heurísticas adaptadas a las particularidades del juego Hex.

2 Hex

El juego de Hex es un fascinante y estratégico juego de mesa en el que dos jugadores compiten para conectar lados opuestos de un tablero hexagonal. A continuación se explica en qué consiste el juego Hex y sus reglas.

2.1 Objetivo del juego

Cada jugador tiene el objetivo de crear un camino continuo de fichas propias que conecte dos lados opuestos de su color. Uno de los jugadores debe conectar los bordes izquierdo y derecho, mientras que el otro debe conectar los bordes superior e inferior.

2.2 Reglas

- Turnos alternos: Los jugadores colocan una ficha en una celda hexagonal vacía del tablero durante su turno.
- Sin movimiento de fichas: Una vez que una ficha se coloca en el tablero, no se puede mover ni eliminar.
- Ganador: El primer jugador que logre conectar sus lados opuestos gana la partida.
- No hay empate: En el juego de Hex, siempre hay un ganador debido a la estructura del tablero y las reglas del juego.

3 Algoritmo Implementado

El algoritmo principal del programa consiste en un Minimax con poda alfa-beta. Este permite explorar el árbol de posibilidades y seleccionar el movimiento que maximiza la posición del jugador mientras minimiza la del adversario. En cada hoja del árbol (o cuando se alcanza la profundidad límite), se evalúa el estado del juego utilizando una función heurística.

4 Heurísticas

4.1 Dijkstra

La heurística base, consiste en una función que calcula el número mínimo de movimientos necesarios para conectar los bordes objetivos. Teniendo en cuenta la siguiente función de peso:

- Llegar a una celda que contiene la ficha del jugador tienen costo 0.

- Llegar a una celda vacía tiene costo 1.
- Una celda ocupada por el adversario se considera bloqueada (coste infinito).

La diferencia entre las distancias mínimas de ambos jugadores (distancia del oponente - dist propia) determina la heurística

4.2 Incorporación de puentes

Se ha modificado el algoritmo Dijkstra para considerar conexiones por puente.

- Cuando dos fichas propias pueden formar puente mediante dos rutas seguras (descompuestas en dos movimientos consecutivos), se le asigna un costo 0 (como si estuvieran adyacentes). Esto “acorta” virtualmente el camino, favoreciendo posiciones con estructuras de puente.

4.3 Evaluación de Puentes y Posiciones Críticas

4.3.1 Detección de puentes

Luego de probar una posición, se verifica si esta forma un puente con otra casilla, y se le suma un bonus adicional.

4.3.2 Verificación de posiciones críticas

Se determina si una celda es crítica dentro de un puente. Una posición crítica es aquella en la que, para completar la conexión, se necesita ocuparla, especialmente si la celda complementaria ya se encuentra ocupada por el adversario.

Dada una posición, identifica si es parte de un puente y chequea si la otra celda crítica está ocupada por el adversario. En ese caso, se otorga un bonus a la evaluación, incentivando a la IA a jugar en esa posición para asegurar o completar el puente.