Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI5437 - Inteligencia Artificial I
Trimestre Enero-Marzo 2025

Connect6

Profesor: Guillermo Palma

Alumnos: Eliezer Cario - Jhonaiker Blanco

Carnés: 18-10605 - 18-101784

Caracas, 22 de Marzo de 2025

Descripción del sistema

El sistema desarrollado es un agente capaz de jugar el juego Connect6 contra un jugador humano en consola. El agente implementa el algoritmo Monte Carlo Tree Search (MCTS), complementado con heurísticas específicas del juego que priorizan jugadas defensivas y ofensivas basadas en patrones críticos del tablero. La representación del juego se encuentra en el módulo board.go, el cual define la estructura del tablero de 19×19, junto con funciones para aplicar movimientos (diferenciando jugadas de una sola ficha en el primer turno mediante un centinela, –1,–1), verificar condiciones de victoria (6 fichas consecutivas en cualquiera de las direcciones) y evaluar el estado mediante EvaluateBoard, que pondera cadenas de fichas según su longitud y bloqueos.

El algoritmo MCTS, implementado en mcts.go, guía la toma de decisiones mediante simulaciones estocásticas. Se construye de forma incremental un árbol de búsqueda en el que se emplea la fórmula UCB (usando un factor de exploración $\sqrt{2}$) para balancear la exploración y explotación de movimientos. La búsqueda se detiene cuando se alcanza un límite de tiempo (tpj), asegurando que el agente realice suficientes simulaciones sin exceder el tiempo asignado. Para optimizar la respuesta ante situaciones críticas, antes de iniciar la simulación MCTS se verifica si existen amenazas inmediatas (tanto jugadas ganadoras para el jugador como para el oponente) mediante funciones auxiliares como FindCriticalBlocks y FindBestComplementForCritical. Esto permite que, en caso de detectar una amenaza inminente, el agente genere un movimiento de bloqueo óptimo, en situaciones ofensivas, encuentre jugadas ganadoras inmediatas usando un pre-filtrado en FindWinningMove con un umbral ajustado en EvaluateBoard.

Esto se realiza a través de WeightedChainScore la cual asigna puntajes basados en la cantidad y calidad de cadenas de fichas lo que permite tanto evaluar el potencial ofensivo como defensivo de un estado y así lograr descartar aquellos estados poco prometedores y centrándose en candidatos que conduzcan a victorias inmediatas.

La interfaz por línea de comandos, definida en ui.go, muestra el tablero en cada turno, solicita la entrada del usuario y valida los movimientos (solicitando una única posición en el primer turno y dos en los siguientes). El módulo game.go coordina el flujo del juego, alternando los turnos entre el jugador humano y el bot (este último usando MCTS para tomar decisiones) y anunciando al ganador al finalizar la partida.

Bibliografía

Magnuson, Max (2015) "Monte Carlo Tree Search and Its Applications," *Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal*: Vol. 2: Iss. 2, Article 4.
Available at: https://digitalcommons.morris.umn.edu/horizons/vol2/iss2/4

michaelbzms. (2021). MonteCarloTreeSearch [Software]. Github.
https://github.com/michaelbzms/MonteCarloTreeSearch