Building a Adversarial Game Playing Agent

Author: Ray Phan

Date: July 2, 2021

Introduction

For this project, I decided to implement an advanced custom technique (Option 3) that was not covered in the lecture, namely the Monte Carlo Tree Search algorithm. I followed the pseudocode and recommendations as suggested in the resources section of the nanodegree. A custom class to represent a Monte Carlo Tree Search node as well as the accompanied functions that facilitate the search and best move recommendation were implemented.

For performance analysis, 100 matches were run with the baseline and the Monte Carlo Tree Search algorithm. For the baseline, the standard Minimax algorithm was chosen and was already available in the project which was selectable by a command-line argument. Fair matches were **not enabled** as they would not have a significant impact given that the agent always selects a random move at first.

Performance of Monte Carlo Tree Search vs. Minimax

The performance analysis provided shows the success rate at various time limits. The table below illustrates the success rate of each algorithm over 100 matches as we increase the time limit for a move.

Time Limit (ms)	Minimax	Monte Carlo Tree Search
10	40%	23%
50	43%	26%
100	46%	27%
150	50%	29%
200	55%	30%
250	62%	33%

Note that there is an improvement in performance as we increase the time limit. This is due to the algorithm having more time to run simulations before ultimately deciding what the best action is. As we can see here, the Monte Carlo Tree Search algorithm does not performwell in comparison to Minimax.

As noted by this Ph.D. thesis, Monte Carlo Tree Search has an advantage over traditional depth-limited minimax search with alpha-beta pruning in games with high branching factors such as Go. However, Minimax search with alpha-beta pruning still surpasses Monte Carlo Tree Search in domains like Chess.

Studies show that Monte Carlo Tree Search does not detect shallow traps, where opponents can win within a few moves, as well as minimax search. Thus, minimax search performs better than Monte Carlo Tree Search in games like Chess, which can end instantly (king is captured). As the Isolation game is very similar to Chess in terms of moves, this could be a factor in its poor performance.