

Build a Game-Playing Agent

Mohan Murugesan

Introduction

In this project, we develop an adversarial search agent to play the game "Isolation". Isolation is a deterministic, two-player game of perfect information in which the players alternate turns moving a single piece from one cell to another on a board. Whenever either player occupies a cell, that cell becomes blocked for the remainder of the game. The first player with no remaining legal moves loses, and the opponent is declared the winner. This project uses a version of Isolation where each agent is restricted to L-shaped movements (like a knight in chess) on a rectangular grid (like a chess or checkerboard).

The main objectives of this project are:

- Implement Minimax algorithm for searching the game tree.

- Implement minimax search with alpha-beta pruning.

- Implement Iterative Deepening along with the Alpha-Beta search that returns the best next move within provided time.

- Develop heuristic evaluation functions

Evaluation Functions

➤ Heuristic 1:

Score calculated first based on the number of moves available to the player, with respect to the available moves to the opponent. Board is penalized, if the opponent has more available moves than the player. Also normalizing the difference in moves with the manhattan distance between the player. Board is penalized, if the manhattan distance between the player and the opponent is large as it will be hard for player to block moves for opponent.

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	10	0	9	1	9	1	9	1
2	MM_Open	7	3	6	4	9	1	6	4
3	MM_Center	9	1	10	0	10	0	8	2
4	MM_Improved	9	1	8	2	9	1	7	3
5	AB_Open	6	4	4	6	6	4	4	6
6	AB_Center	5	5	5	5	3	7	7	3
7	AB_Improved	7	3	7	3	3	7	6	4
Win Rate:		75.7%		70.0%		70.0%		67.1%	

➤ Heuristic 2:

Score calculated based on player current state of the board, penalizing player for moving to corner. As corner position decreases number of available moves, player penalized for corner

positions. This is done after calculating number of legal moves for player compared to opponent.

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	10	0	10	0	10	0
2	MM_Open	7	3	8	2	7	3	5	5
3	MM_Center	8	2	10	0	10	0	9	1
4	MM_Improved	6	4	5	5	6	4	5	5
5	AB_Open	3	7	5	5	5	5	6	4
6	AB_Center	6	4	4	6	8	2	6	4
7	AB_Improved	5	5	3	7	4	6	6	4
Win Rate:		62.9%		64.3%		71.4%		67.1%	

➤ **Heuristic 3:**

Score calculated based on the distance between the player and the opponent, It prefers maximum distance i.e. run away from the opponent. Returns the absolute difference between the sums of the location vectors, where larger differences equal higher scores.

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	8	2	9	1	10	0
2	MM_Open	7	3	7	3	7	3	7	3
3	MM_Center	7	3	9	1	8	2	8	2
4	MM_Improved	9	1	5	5	7	3	7	3
5	AB_Open	5	5	5	5	5	5	2	8
6	AB_Center	5	5	6	4	7	3	6	4
7	AB_Improved	6	4	1	9	5	5	5	5
Win Rate:		68.6%		58.6%		68.6%		64.3%	

Based on the tournament result of all three heuristics, Heuristic 1 is recommended to be used.

1. Out of all heuristics tested in the tournament, heuristic 1 was the one which performed best.
2. Heuristic 1 is based on the number of moves available and distance between player & opponent. This is important in as it will be difficult for player to block opponent, if distance is more between player and opponent.