BUILD AN ISOLATION GAME PLAYING AGENT :

1 INTRODUCTION

In this project, we developed an AI agent for playing the game of Isolation which is a two-player game played on a rectangular grid of positions where both players take turns to move their piece and the first player who runs out of available moves loses the game.

The main objectives of this project are:

- Implement Minimax algorithm for searching the game tree.
- Implement Alpha-Beta pruning
- Develop heuristic evaluation functions

2 HEURISTIC EVALUATION FUNCTIONS

- Heursitic-One: len(my available moves) – α len(available opponent moves) (α=1.5)
  The role of this heuristic is to maximize opponent moves.
- Heuristic-two:
  improved version of heuristic three where we are returning higher gain after
  calling heuristic-three in Case we have a center location move (based on the lecture player who got the center locations usually win the game)

- Heuristic-three:

  $len(my\ available\ moves)$*$len(my\ available\ moves)$
  $-\alpha\ len(available\ opponent\ moves)$*$len(available$ opponent $moves)$
  with α chosen as 1.5. (I played with different values for α but the optimal one was 1.5)

Running the tournament on 5 matches produce the below data

Match	Opponent	AB_IMPROVED (won,lost)	AB_CUSTOM (won,lost)	AB_CUSTOM_2 (won,lost)	AB_CUSTOM_3 (won,lost)	
1	Random	8,2	8,2	7,3	9,1	
2	MM_open	5,5	5,5	8,2	6,4	
3	MM_center	9,1	8,2	6,4	8,2	
4	MM_Improve	6,4	3,7	5,5	8,2	
5	AB_Open	4,6	4,6	5,5	4,6	
6	AB_Center	6,4	6,4	6,4	7,3	
7	AB_Improved	5,5	3,7	5,5	6,4	
8	Win Rate	61%	52%	60%	68%	

From the above diagram we can conclude that custom_3 provide the higher rate percentage using the α=1.5

Heuristic Choice: AB_CUSTOM_3

- Provide a highest score than the other heuristics
- Running time complexity is comparable to the improved
  Heuristic so it should not affect the maximum depth searched
- Room for improvement if we can combine the custom_3 with the custom_2 heuristics