PROJECT REPORT

ADVERSARIAL SEARCH

Keshav Musunuri

OPTION 1: ADVANCED HEURISTIC

I have chosen the option one for this project. The advanced heuristic I am using for this problem is twofold. Firstly if the current position of the player(we play) is not at the border of the board I am using the *super aggressive improved score (SAIS)* taken from this article on medium 'https://medium.freecodecamp.org/playing-strategy-games-with-minimax-4ecb83b39b4b' . It is a simple linear combination of the players and the opponents moves, penalizing heavily on the remaining opponent moves with a weight of '-3'. On the other hand if the current position of the player is at the border of the board then I am not only penalizing the current moves/liberties of the opponent but also penalizing on the moves that could be taken next after every move taken now with a factor of '-3', the code for this implementation can be found on my_custom_player.py. This kind of heuristic is not only promising but also effective in generating desired results as we are not only cornering the opponent in general but also heavily imposing over the opponents moves in the case that the players situation itself is in doubt(at border). I played 100 games against the minimax agent and the greedy agent which are more competitive opponents and these are the results generated.

RESULTS:

	Opponent	No of Matches	Time limit	Custom player win %
Basic heuristic				
	MiniMax	100	150s	50
	MiniMax	100	300s	50
	Greedy	100	150s	75
Advanced heuristic				
	MiniMax	100	150s	84
	MiniMax	100	300s	82
	Greedy	100	150s	87.5

QUESTIONS:

What features of the game does your heuristic incorporate, and why do you think those features matter in evaluating states during search?

My heuristic incorporates the *motto* of the game to limit the opponents moves. This is very important and matter a lot in the succeeding in this game as it is the main target of the game and the better way you try to limit the number of moves for the opponent the better the chances to win, at least that's what I learned repeatedly playing this game. In order to achieve this, as I explained earlier I am trying to penalize the opponents moves heavily especially in the situation when our player is in a critical situation like in the border

Analyze the search depth your agent achieves using your custom heuristic. Does search speed matter more or less than accuracy to the performance of your heuristic?

In my observation during search I noticed an increasing trend between depth and the ply_count Indicating that the custom heuristic is more inclined towards deeper search in an attempt to win the game. With respect to search speed or accuracy matters more to the performance, as deeper search is generally achieved near the end of the game it seems that speed might matter more. We could also notice that in the results generated above.