

Heuristic analysis

For this project, I try to create three different strategies to evaluate the better steps for computer players, and then mixed those strategies to find an average result.

Heuristic 1

The first strategy is to calculate the numbers of moves for players, and then subtract the numbers of opponent's moves in order to find a score evaluate which move is better. If there is a score higher than another, we can know that this move is able to have higher chance to win.

```
own_moves_score = len(game.get_legal_moves(player))
opp_moves_score = len(game.get_legal_moves(game.get_opponent(player)))

return float(own_moves_score - opp_moves_score)
```

Heuristic 2

I add a new criterion called center score to help to analyze the better move. This criterion is to calculate the distance between center and the move we concerned. We can use this method to compare the distance of player to opponent. We can guess if the move is closer to center it should be considered a better move.

```
center_y_pos, center_x_pos = int(game.height / 2), int(game.width / 2)
player_y_pos, player_x_pos = game.get_player_location(player)
opponent_y_pos, opponent_x_pos = game.get_player_location(game.get_opponent(player))
player_distance = abs(player_y_pos - center_y_pos) + abs(player_x_pos - center_x_pos)
opponent_distance = abs(opponent_y_pos - center_y_pos) + abs(opponent_x_pos - center_x_pos)
center_score = float(opponent_distance - player_distance)

return float(own_moves_score - opp_moves_score + center_score)
```

Heuristic 3

The third heuristic includes another criterion called common-score which is able to calculate the intersection between player and opponent. I assume that the more common move between player and opponent is a better move because the player has more opportunities to interfere opponent. Thus, in this heuristic I combined heuristic1,2 and 3 to retrieve an average result.

```
common_score = float (len(set(own_moves).intersection(set(opp_moves))))
```

Tournament Results

Playing Matches

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	8	2	9	1	8	2	9	1
2	MM_Open	6	4	7	3	8	2	7	3
3	MM_Center	5	5	8	2	7	3	7	3
4	MM_Improved	7	3	8	2	6	4	6	4
5	AB_Open	6	4	5	5	7	3	7	3
6	AB_Center	5	5	4	6	5	5	5	5
7	AB_Improved	5	5	4	6	3	7	5	5
Win Rate:		60.0%		64.3%		62.9%		65.7%	

Playing Matches

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	8	2	10	0	8	2	9	1
2	MM_Open	6	4	6	4	9	1	6	4
3	MM_Center	6	4	7	3	7	3	9	1
4	MM_Improved	4	6	7	3	7	3	7	3
5	AB_Open	6	4	6	4	5	5	4	6
6	AB_Center	4	6	5	5	7	3	5	5
7	AB_Improved	5	5	5	5	5	5	4	6
Win Rate:		55.7%		65.7%		68.6%		62.9%	

For the match 3 and 6, the AB_Custom_2 always perform better result than AB_Improved and AB_Custom because the it's score is influenced by the center heuristic function. It means that when the board game is close to the center, the AB_Custom_2 will obtain greater chance to win. In addition, although I mixed heuristic 1 & 2 to the AB_Custom_3 it does not always play best result in every match. It is because the heuristic center score and common score do not always influence efficiently to the win or lose, but sometime they can be very effective to help us to choose the best step. Thus, when using heuristic 3 we can always retrieve average result but not best result.

Recommendations

Based on the results in the tournament of all the three heuristics, I would be inclined to recommend heuristic 3 to be used because of the following reasons:

- Heuristic 3 can perform average or even better result in different situation of game broad because it mixed two heuristics in one.

- Heuristic 3 includes positional advantage to evaluate the better step in game board by distance measurement.
- Heuristic 3 is able to block available move to the opponent, and always selecting board states where there is a possibility for the player to block the opponent. Also, it does increase complexity of programming and then we can get better result.