Heuristic Analysis

Adam Liu

As instructed in the project material, I have written 3 custome_score functions in order to evaluate the heuristic value of a game state from the point of view of the given player. The basic concept of those 3 function came from the previous course material (#my_moves - #opponent_moves), and was customised and improved in certain ways.

Function custom_score:

This function is the best heuristic function for my project submission. This function is generally the combination of custom_score_2 & custom_score_3 (for the explanations of custom_score_2 & custom_score_3 functions please see the following parts). This function has the highest performance.

Function custom_score_2:

This function makes the player aggressive due to the reason it only calculates advantage the player has based on how far ahead or how far behind by comparing the moves that the player has and the player's opponent has. In addition to this, I also applied power to the function, which enhances the advantages the player has. With this calculation implemented, the player would always try to maximise the returned value, which means it would always try to reduce the opponent's moves while the it has the most possible moves.

Function custom_score_3:

This function is relatively similar to the second function, the only difference is that instead of enhancing the player's remaining moves, the function enhances the opponent's advantages so the player would be much less aggressive.

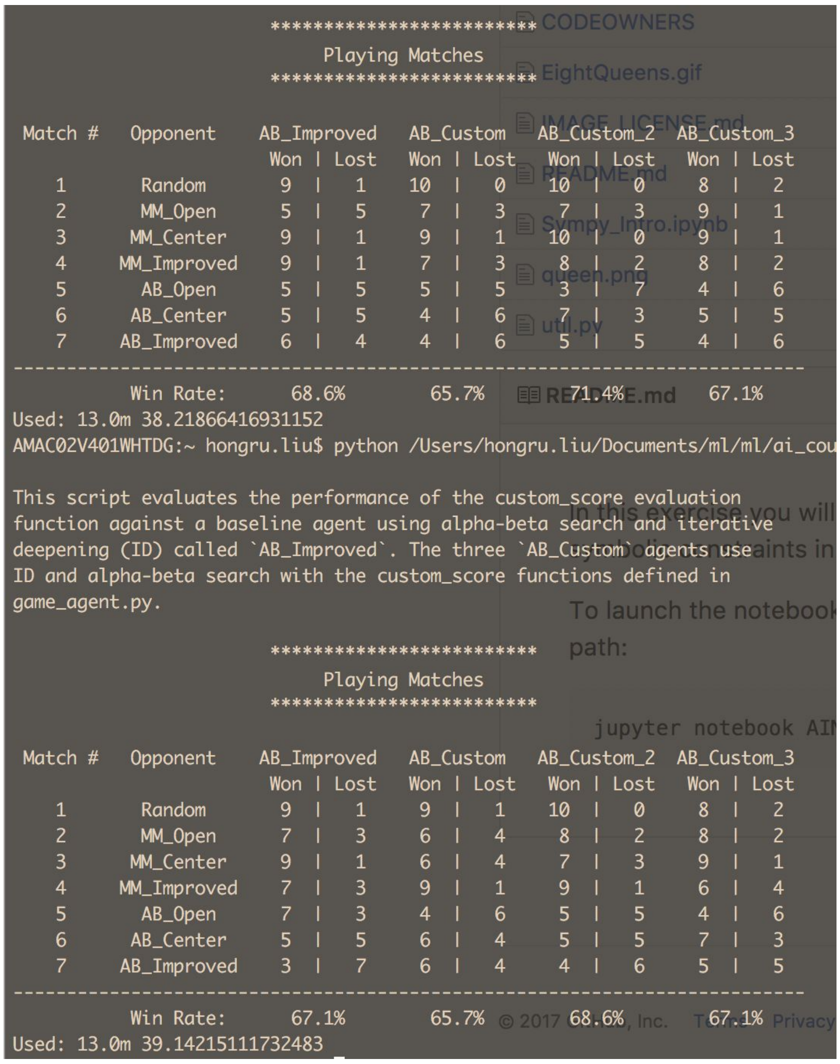
Conclusion:

The custom_score function was actually the combination of the custom_score_2 & custom_score_3 functions. Custom_score_2 would give the player a stronger sense of the advantage the player has, which led to better results, and the sense of advantage would be much less if we use custom_score_3, and the results of using this heuristic are less impressive than custom_score_2. However, custom_score had the worst results among all three, due to the reason it is a combination of both enhancing the advantage and reducing the advantage, the heuristic ended up in the middle, giving much vaguer sense of the situation.

I would recommend the custom_score_2 heuristic for the following reasons:

1. It increases the advantages the player has for the player to better weight the situations
2. Using an exponential representation of the advantage, delivers the measurements without increasing the computation requirements.
3. It derives from simple intuition that human player would use to play.

Please find performance screenshots in the following pages:



```
                          ************************
                               Playing Matches
                          ************************

 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     8  |   2
    2       MM_Open      5  |   5     7  |   3     7  |   3     9  |   1
    3      MM_Center     9  |   1     9  |   1    10  |   0     9  |   1
    4     MM_Improved    9  |   1     7  |   3     8  |   2     8  |   2
    5       AB_Open      5  |   5     5  |   5     3  |   7     4  |   6
    6      AB_Center     5  |   5     4  |   6     7  |   3     5  |   5
    7     AB_Improved    6  |   4     4  |   6     5  |   5     4  |   6
--------------------------------------------------------------------------
           Win Rate:      68.6%        65.7%        71.4%        67.1%
Used: 13.0m 38.21866416931152
AMAC02V401WHTDG:~ hongru.liu$ python /Users/hongru.liu/Documents/ml/ml/ai_cou

This script evaluates the performance of the custom_score evaluation
function against a baseline agent using alpha-beta search and iterative
deepening (ID) called `AB_Improved`. The three `AB_Custom` agents use
ID and alpha-beta search with the custom_score functions defined in
game_agent.py.

                          ************************
                               Playing Matches
                          ************************

 Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                        Won | Lost   Won | Lost   Won | Lost   Won | Lost
    1       Random       9  |   1     9  |   1    10  |   0     8  |   2
    2       MM_Open      7  |   3     6  |   4     8  |   2     8  |   2
    3      MM_Center     9  |   1     6  |   4     7  |   3     9  |   1
    4     MM_Improved    7  |   3     9  |   1     9  |   1     6  |   4
    5       AB_Open      7  |   3     4  |   6     5  |   5     4  |   6
    6      AB_Center     5  |   5     6  |   4     5  |   5     7  |   3
    7     AB_Improved    3  |   7     6  |   4     4  |   6     5  |   5
--------------------------------------------------------------------------
           Win Rate:      67.1%        65.7%        68.6%        67.1%
Used: 13.0m 39.14215111732483
```