Heuristic

November 21, 2017

0.0.1 heuristic_analysis

Your analysis should conclude with a comparison of the different heuristics and your reasoning for choosing the heuristic you ultimately use in your submitted agent.

- 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.

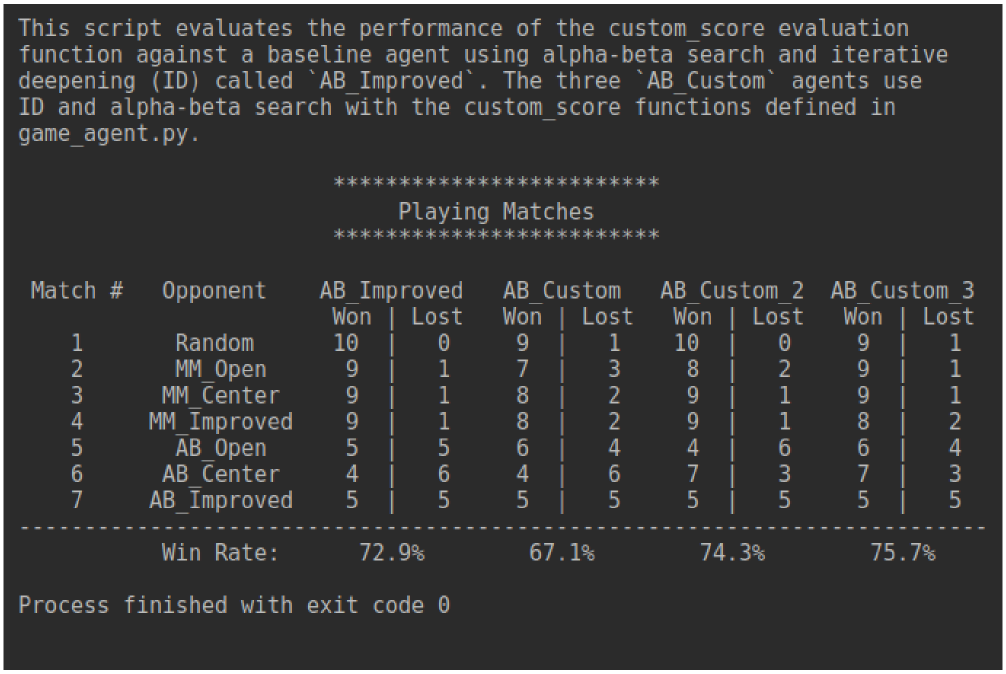
Each code Calculate the heuristic value of a game state from the point of view of the given player.

```
def custom_score(game, player):
    loser_winner(player,game)
    player_move = len(game.get_legal_moves(player))
    opponent_moves = len(game.get_legal_moves(game.get_opponent(player)))
    blank_spaces = len(game.get_blank_spaces())
    return float((blank_spaces-opponent_moves)*player_move-opponent_moves)


def custom_score_2(game, player):
    loser_winner(player,game)
    my_moves = len(game.get_legal_moves(player))
    opponent_moves = len(game.get_legal_moves(game.get_opponent(player)))
    blank_spaces = len(game.get_blank_spaces())
    return float(my_moves*(blank_spaces-1) - opponent_moves*(blank_spaces))


def custom_score_3(game, player):
    loser_winner(player,game)
    my_moves = len(game.get_legal_moves(player))
    opponent_moves = len(game.get_legal_moves(game.get_opponent(player)))
    return float(my_moves - 3*opponent_moves)
```

- custom_score is only focus on reducing opponent's move, which is fuinding the gap between player_moves and opponent moves



```
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      10  |   0     9  |   1    10  |   0     9  |   1
    2       MM_Open      9  |   1     7  |   3     8  |   2     9  |   1
    3      MM_Center     9  |   1     8  |   2     9  |   1     9  |   1
    4     MM_Improved    9  |   1     8  |   2     9  |   1     8  |   2
    5       AB_Open      5  |   5     6  |   4     4  |   6     6  |   4
    6      AB_Center     4  |   6     4  |   6     7  |   3     7  |   3
    7     AB_Improved    5  |   5     5  |   5     5  |   5     5  |   5
--------------------------------------------------------------------------
           Win Rate:      72.9%        67.1%        74.3%        75.7%

Process finished with exit code 0
```

result

- custom_score_2 is focused to increase my_moves.
- custom_score_3 is focused to decrease opponent_moves.

0.1 Tournament Results

Winner: AB_Custom_3 has the highest win rate: 75.7% followed by AB_Custom_2 with 74.3%

we can see custome_2 has better start ar Random with 10 wins and no lost, however it is custom_3 which came back to the game and wins most of the games.

also we can expect to have harder opponent in AB games (Alpha Beta) in comare to MM (Min Max) and Random, that explains the reason we are having less wins in last 3 games.

we have high similarity between all functions except AB_Custom which has the lowest performance

0.1.1 Reasoning for choosing AB_Custom_3:

1. Highest Overal **Winning Rate**
2. depth: because opp_moves is considered,it is one layer deeper than open_score for the same depth limitation.
3. Less Complex in compare to other cutome model.