Heuristic analysis

Fangjun Shi

Evaluation heuristics functions analyze

Evaluation heuristics 1

```
if game.is_loser(player):
        return float("-inf")
if game.is_winner(player):
        return float("inf")
blank_spaces = len(game.get_blank_spaces())
own_moves = len(game.get_legal_moves(player))
opp_moves = len(game.get_legal_moves(game.get_opponent(player)))
return float(blank_spaces / 3 + own_moves / 3  - opp_moves * 3)
```

This function imitate *improved_score( )* in *sample_players.py*,.The legal moves number is used to make sure that have enough the number of common moves and the opponent of the specified playerand, add blank spaces to improve the number of moves available. This is a conservative defensive heuristic.

Evaluation heuristics 2

```
if game.is_loser(player):
        return float("-inf")
if game.is_winner(player):
        return float("inf")
own_moves = len(game.get_legal_moves(player))
opp_moves = len(game.get_legal_moves(game.get_opponent(player)))
w, h = game.width / 2., game.height / 2.
y, x = game.get_player_location(player)
return float(blank_spaces / 2  - opp_moves * 2 + ((h - y) ** 2 + (w - x)
** 2) / 4)
```

This function imitate *center_score( ) and improved_score( )* in *sample_players.py*, it considers the distance from the center of the board to the position of the player. Defense in attack at the same time. So, this is an eclectic heuristic.

Evaluation heuristics 3

```
if game.is_loser(player):
        return float("-inf")
if game.is_winner(player):
        return float("inf")
blank_spaces = len(game.get_blank_spaces())
own_moves = len(game.get_legal_moves(player))
opp_moves = len(game.get_legal_moves(game.get_opponent(player)))
w, h = game.width / 2., game.height / 2.
y, x = game.get_player_location(player)
return float(blank_spaces / 4 +  own_moves / 4 - opp_moves * 4 + ((h - y)
** 2 + (w - x) ** 2) / 4)
```

This function is based on the last function, and use blank spaces number. This is also an eclectic heuristic but more careful than the last.

Performance

```
                         *************************
                              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      7   |   3      8   |   2     10   |   0
    2         MM_Open     5   |   5      6   |   4      6   |   4     10   |   0
    3        MM_Center    7   |   3      9   |   1      9   |   1      8   |   2
    4      MM_Improved    8   |   2      6   |   4      7   |   3      7   |   3
    5         AB_Open     6   |   4      4   |   6      4   |   6      6   |   4
    6        AB_Center    7   |   3      6   |   4      5   |   5      7   |   3
    7      AB_Improved    4   |   6      3   |   7      6   |   4      5   |   5
----------------------------------------------------------------------------------
            Win Rate:      65.7%        58.6%        64.3%        75.7%
```

In this three function, as you can see, the third one (AB_Custom_3) performed the best (won 75.7%). In the first match, they are all looks good, but the third wining 10 out of 10 games; and in the fifth match, they are not looks good, but the third's win rate is the best one than others.With the match getting difficulty, the win rate gets lower, and the third one's win rate is always equal or greater than others. The first is unaggressive, and second is a little careless. Maybe a good heuristics should attack, defend and more careful.

Maybe there are other ways can improve the win rate.

Recommendation

In summary, the third evaluation function (AB_Custom_3) should be used.
Here are three reasons:

1. AB_Custom_3 has the best final win rate than others, and each match win rate are equal or greater than 50%.
2. The code for AB_Custom_3 is more complicated. AB_Custom_3 almost contains AB_Custom_1 and AB_Custom_2.
3. AB_Custom_3 is the most comprehensive that considers the legal moves number, blank spaces and he distance from the center of the board to the position.