

Evaluation function Heuristic report:

The project required an analysis of different kinds of heuristics used for evaluation functions. I've tried three different heuristics and compared their performances:

1. **custom_score_2:**

This function returns a higher scores for moves that employ an aggressive game tactic. It has the effect of chasing and trying to corner the opponent. During the search it returns $(\text{own_moves} - 3 * \text{opponent_moves})$

CODE:

```
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)))
return float(- opp_moves)
```

2. **custom_score_3:**

This function penalizes those moves that are nearer to the walls of the board, since those positions restrict the number of possible moves that can be made. This becomes important as the game progresses.

CODE:

```
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)))
return float(2*own_moves - opp_moves)
```

3. **custom_score:**

This function combines the techniques involved in the above two functions by using them and adjusting the relative importance of each.

CODE:

```
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)))
return float(own_moves - 2*opp_moves)
```

Based on the heuristics, these are the respective scores and their relative performances:

```
smacar@Smacar-U-LT-006:~/Desktop/opencv/AI/similar$ python tournament.py

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         9 | 1         7 | 3
2         MM_Open     6 | 4         5 | 5         7 | 3         5 | 5
3         MM_Center    7 | 3         8 | 2         7 | 3         8 | 2
4         MM_Improved  6 | 4         6 | 4         5 | 5         5 | 5
5         AB_Open      6 | 4         8 | 2         8 | 2         4 | 6
6         AB_Center    7 | 3         4 | 6         5 | 5         5 | 5
7         AB_Improved  6 | 4         7 | 3         6 | 4         5 | 5
-----
Win Rate:   67.0%       67.0%       67.0%       55.0%
```

COMPARING THE STUDENTS AND ID_IMPROVED :

```
*****
Evaluating: ID_Improved
*****

Playing Matches:
-----
Match 1: ID_Improved vs Random    Result: 1724 to 276
Match 2: ID_Improved vs MM_Null   Result: 1395 to 605
Match 3: ID_Improved vs MM_Open   Result: 1012 to 988
Match 4: ID_Improved vs MM_Improved Result: 952 to 1048
Match 5: ID_Improved vs AB_Null   Result: 1297 to 703
Match 6: ID_Improved vs AB_Open   Result: 1150 to 850
Match 7: ID_Improved vs AB_Improved Result: 1142 to 858

Results:
-----
ID_Improved      61.94%

*****
Evaluating: Student1
*****

Playing Matches:
-----
Match 1: Student1 vs Random    Result: 1736 to 264
Match 2: Student1 vs MM_Null   Result: 1458 to 542
Match 3: Student1 vs MM_Open   Result: 1136 to 864
Match 4: Student1 vs MM_Improved Result: 1036 to 964
Match 5: Student1 vs AB_Null   Result: 1411 to 589
Match 6: Student1 vs AB_Open   Result: 1230 to 770
Match 7: Student1 vs AB_Improved Result: 1218 to 782

Results:
-----
Student1         65.89%
```

```
Evaluating: Student2
*****
```

```
Playing Matches:
```

```
-----
Match 1: Student2 vs Random Result: 1745 to 255
Match 2: Student2 vs MM_Null Result: 1433 to 567
Match 3: Student2 vs MM_Open Result: 1106 to 894
Match 4: Student2 vs MM_Improved Result: 1041 to 959
Match 5: Student2 vs AB_Null Result: 1388 to 612
Match 6: Student2 vs AB_Open Result: 1235 to 765
Match 7: Student2 vs AB_Improved Result: 1209 to 791
```

```
Results:
```

```
-----
Student2 65.41%
```

```
*****
Evaluating: Student3
*****
```

```
Playing Matches:
```

```
-----
Match 1: Student3 vs Random Result: 1724 to 276
Match 2: Student3 vs MM_Null Result: 1412 to 588
Match 3: Student3 vs MM_Open Result: 1092 to 908
Match 4: Student3 vs MM_Improved Result: 1018 to 982
Match 5: Student3 vs AB_Null Result: 1336 to 664
Match 6: Student3 vs AB_Open Result: 1210 to 790
Match 7: Student3 vs AB_Improved Result: 1197 to 803
```

```
Results:
```

```
-----
Student3 64.21%
```

BEST HEURISTIC:

custom_score_2, the one employing aggressive gameplay has the most performance gain, over other heuristics Because it employs the technique of cornering the opponent, thus reducing the number of moves for the opponent.

This technique also allows the game to end earlier since more number of boxes are left on the board while the opponent is pushed to a corner.