HEURISMIC ANALYS

Build a Game-Playing Agent

Antonio Rincon

Contents

[1. Base Heuristics 3](#_Toc502765344)

[2. Help Functions 3](#_Toc502765345)

[3. Custom Heuristic 4](#_Toc502765346)

[4. Evaluation 6](#_Toc502765347)

# Base Heuristics

In the starting code provided with the isolation project there are several heuristics provided:

* **open\_move\_score**

score equal to the number of moves open for your computer player on the board.

* **null\_score**

This heuristic presumes no knowledge for non-terminal states, and returns the same uninformative value for all other states

* **center\_score**

Outputs a score equal to square of the distance from the center of the board to the position of the player

* **improved\_score**

outputs a score equal to the difference in the number of moves available to the two players.

# Help Functions

As part to beat the improved score, new functions have been added to the code:

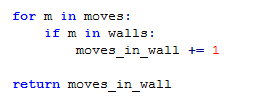
* **improved\_score**

Revision of the improved\_score but, outputs a score equal to the difference in the number of moves available to the two players, weighting the opponent moves



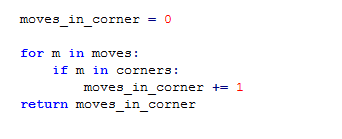
* **in\_wall**

score the presence of move in the cells along the 4 wall borders



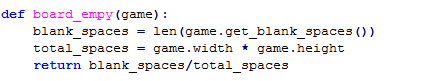
* **in\_border**

score the presence of moves in the 4 corners:



* **board\_empty**

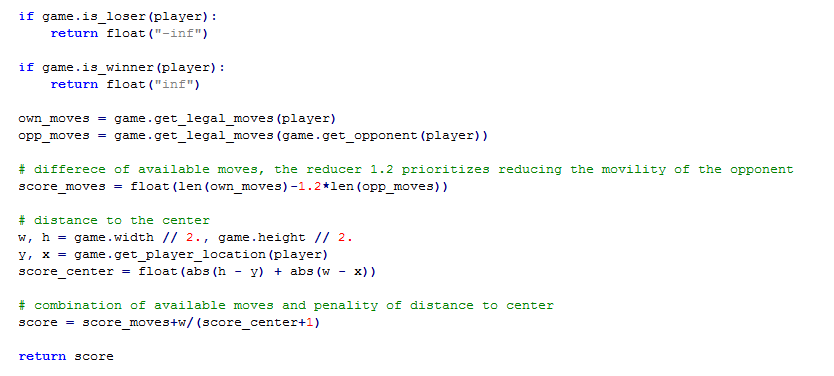
it calculates the percentage of empty cells



# Custom Heuristic

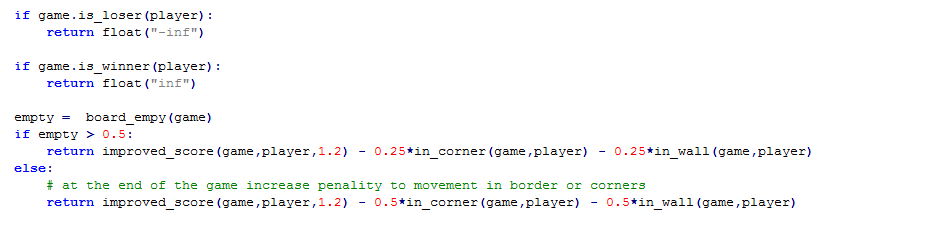
* **custom\_score**

It’s a combination of improved\_score weigthing the opponent moves and penalizing distance to the center. I have tested with different weights, finally I decided to use a weight 1.2 to try to minimize the opponent mobility. Adding a penalty to moves far form center will try to avoid walls and corners



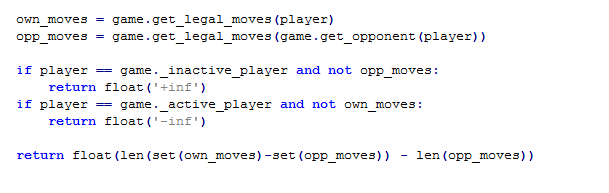
* **custom\_score\_2**

It’s a combination of improved\_score with weight 1.2 and penalizing moves in walls and corner. As the game progress and less cells are empty I have increase the penalty of moves in the walls and corners



* **custom\_score\_3**

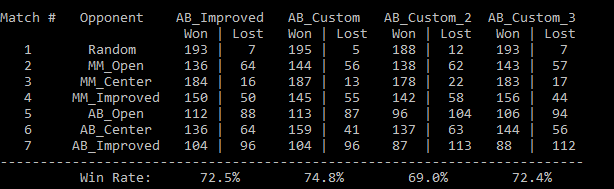
This herustic it’s a revision of improved score, but taking into consideration overlapping moves with the opponent, only moves that are not available to the opponent are taken into consideration to count own movements

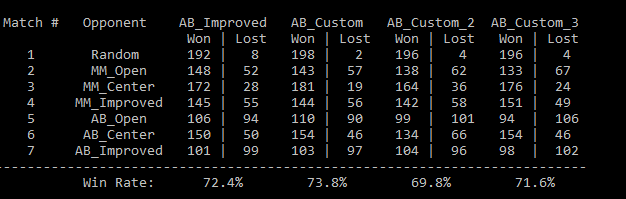


# Evaluation

While testing the tournament application, some warning related to timeouts and forfeit raised. Initialization of best move was changed to selected any available move instead of (-1,-1) to avoid the forfeit warnings. The timeout warning was related to the sleeping mode of the computer.

The was a big variability on the results in different executions, the number of matches per player was increased to 100 to have more homogeneous results in different executions.





It’s clear form these results that the improved\_score and alphabeta algorithm beats minimax and other scores functions used by alphabeta.

From the custom heuristics, it looks like that avoiding wall and corners combining the different of available moves and penalizing moves away from the center provides an small advantage compared to the improved\_score.