

Othello Bot Development Report

Search Strategies Used:

- Performed Alpha-Beta pruning
- A Search algorithm seeks to decrease the number of nodes that are evaluated by the minimax algorithm in its search tree.
- It stops evaluating a move when at least one possibility has been found that proves the move to be worse than a previously examined move. It essentially prunes such nodes in the search tree.
- The tree depth used is 6 to ensure the timeout does not occur.

Heuristics used:

- Heuristic v1
 - The heuristic used here is the difference between player coins and opponent coins.
 - Number of MAX Coins - Number of MIN Coins
- Heuristic v2
 1. Four Factors are considered in this heuristic.
 - **Coin Parity** (Similar to Heuristic v1)
 $((\text{MAX Coins} - \text{MIN coins}) / (\text{Total Coins})) * 100$
 - **Mobility** (Number of Valid Moves)
 $((\text{MAX Moves} - \text{MIN Moves}) / (\text{Total Moves})) * 100$
 - **Corners Captured**
Since Corner coins can't be altered and they can help in capturing more coins.
 $((\text{MAX Corners} - \text{MIN Corners}) / (\text{Total Corners})) * 100$
 - **Stability**
Number of coins that are hard to capture. Stability is expressed as a Weighted Matrix
Let $\text{SUM}(\text{Player})$ be the Weighted Sum of Player coins.
 $((\text{SUM}(\text{MAX}) - \text{SUM}(\text{MIN})) / (\text{SUM}(\text{MAX}) + \text{SUM}(\text{MIN}))) * 100$
 2. The above factors are proved to be useful experimentally in this [research paper](#) . Final heuristic function is the weighted sum of the above mentioned factors. We took the optimal weights given in the paper and tuned them to get better results. Our weights are 0.25, 0.05, 0.3, 0.4 respectively