

Group Members:

-Hasan Pekedis 150120068

-Cem Eren Kula 150120059

-Hasan Ozeren 150121036

Design Document for Othello AI

In this document the AI player utilizes the minimax algorithm with three distinct heuristic evaluation methods (h1, h2, h3) to determine the optimal moves.

Classes and Descriptions

Othello Class:

Manages the game board and rules.

Includes methods for:

- Initializing the board.
- Checking valid moves.
- Applying moves and flipping pieces.
- Printing the board.
- Determining if a player has valid moves.
- Displaying the game result.

AIplayer Class:

The AIplayer class encapsulates the logic for the AI's decision-making process. It includes fields and methods essential for evaluating the game state and selecting the best possible move.

Fields

- **depth:** (int) Represents the depth of the minimax algorithm. A depth of 0 implies no further lookahead.
- **heuristic:** (str) Specifies the heuristic function to be used ("h1", "h2", or "h3").
- **player:** (str) The symbol representing the AI player (e.g., 'X' or 'O').
- **opponent:** (str) The symbol representing the opponent player, determined based on the AI player's symbol.

Methods

- **__init__(self, player):** Initializes the AI player with the given symbol and sets up the opponent symbol.
- **decide_move(self, game):** Determines the best move using the minimax algorithm and returns it in a formatted string (e.g., "2b").
- **heuristic_h1(self, game):** Evaluates the game state by comparing the number of AI and opponent pieces on the board.
- **heuristic_h2(self, game):** Evaluates the game state using a weight matrix to assign scores to board positions.
- **heuristic_h3(self, game):** Evaluates the game state based on the difference in the number of valid moves available to the AI and the opponent.
- **minimax(self, game, depth, maximizing_player, heuristic):** Implements the minimax algorithm to recursively evaluate possible moves and select the optimal one.

Heuristic Evaluation Methods

Heuristic h1: Piece Count Difference

We implemented a heuristic that calculates the difference between the number of pieces belonging to the AI player and the opponent on the board.

Strengths:

- Simple and computationally inexpensive.
- Provides a general sense of dominance on the board.

Weaknesses:

- Ignores positional advantages and mobility.
- May not reflect long-term strategic advantages.

Heuristic h2: Weighted Board Positions

We implemented a heuristic that assigns a weight to each board position based on its strategic value, using a predefined weight matrix.

Weight Matrix:

```
[100, 25, 10, 5, 5, 10, 25, 100],
[25, 25, 2, 2, 2, 2, 25, 25],
[10, 2, 5, 1, 1, 5, 2, 10],
[5, 2, 1, 2, 2, 1, 2, 5],
[5, 2, 1, 2, 2, 1, 2, 5],
[10, 2, 5, 1, 1, 5, 2, 10],
[25, 25, 2, 2, 2, 2, 25, 25],
[100, 25, 10, 5, 5, 10, 25, 100]
```

Strengths:

- Considers positional advantages.
- Rewards control of critical positions like corners.

Weaknesses:

- Requires more computation compared to h1.
- Effectiveness depends on the accuracy of the weight matrix.

Heuristic h3: Mobility Difference

This heuristic evaluates the difference in the number of valid moves available to the AI player and the opponent.

Formula:**Strengths:**

- Reflects the strategic importance of mobility.
- Encourages moves that limit the opponent's options.

Weaknesses:

- Can be misleading in situations where reducing opponent mobility sacrifices key positions.

Performance Analysis**Maximum Ply Number**

The maximum ply number depends on the board size and the depth specified for the minimax algorithm. For an 8x8 board:

- **Maximum Ply:** 5

Time Required

The time required to find the best move depends on the depth and the complexity of the board state and heuristic used to find moves. As an example we made benchmarks on different heuristic against the random player (select all moves randomly). Results are below:

```

Benchmark Results:
=====
Heuristic: h1, Depth: 1, AI Wins: 62, Random Wins: 37, Draws: 1, Avg Time: 0.01 seconds, Accuracy: 62.00%
Heuristic: h1, Depth: 2, AI Wins: 74, Random Wins: 26, Draws: 0, Avg Time: 0.02 seconds, Accuracy: 74.00%
Heuristic: h1, Depth: 3, AI Wins: 76, Random Wins: 20, Draws: 4, Avg Time: 0.12 seconds, Accuracy: 76.00%
=====
Heuristic: h2, Depth: 1, AI Wins: 73, Random Wins: 25, Draws: 2, Avg Time: 0.01 seconds, Accuracy: 73.00%
Heuristic: h2, Depth: 2, AI Wins: 77, Random Wins: 18, Draws: 5, Avg Time: 0.03 seconds, Accuracy: 77.00%
Heuristic: h2, Depth: 3, AI Wins: 87, Random Wins: 11, Draws: 2, Avg Time: 0.21 seconds, Accuracy: 87.00%
=====
Heuristic: h3, Depth: 1, AI Wins: 74, Random Wins: 25, Draws: 1, Avg Time: 0.03 seconds, Accuracy: 74.00%
Heuristic: h3, Depth: 2, AI Wins: 66, Random Wins: 33, Draws: 1, Avg Time: 0.14 seconds, Accuracy: 66.00%
Heuristic: h3, Depth: 3, AI Wins: 82, Random Wins: 17, Draws: 1, Avg Time: 1.70 seconds, Accuracy: 82.00%
=====

```

```

Benchmark Results:
=====
Heuristic: h1, Depth: 1, AI Wins: 59, Random Wins: 34, Draws: 7, Avg Time: 0.01 seconds, Accuracy: 59.00%
Heuristic: h1, Depth: 2, AI Wins: 76, Random Wins: 19, Draws: 5, Avg Time: 0.02 seconds, Accuracy: 76.00%
Heuristic: h1, Depth: 3, AI Wins: 78, Random Wins: 21, Draws: 1, Avg Time: 0.11 seconds, Accuracy: 78.00%
Heuristic: h1, Depth: 4, AI Wins: 85, Random Wins: 11, Draws: 4, Avg Time: 0.90 seconds, Accuracy: 85.00%
Heuristic: h1, Depth: 5, AI Wins: 88, Random Wins: 11, Draws: 1, Avg Time: 9.01 seconds, Accuracy: 88.00%
=====
Heuristic: h2, Depth: 1, AI Wins: 72, Random Wins: 24, Draws: 4, Avg Time: 0.01 seconds, Accuracy: 72.00%
Heuristic: h2, Depth: 2, AI Wins: 88, Random Wins: 11, Draws: 1, Avg Time: 0.03 seconds, Accuracy: 88.00%
Heuristic: h2, Depth: 3, AI Wins: 92, Random Wins: 7, Draws: 1, Avg Time: 0.22 seconds, Accuracy: 92.00%
Heuristic: h2, Depth: 4, AI Wins: 88, Random Wins: 10, Draws: 2, Avg Time: 2.14 seconds, Accuracy: 88.00%
Heuristic: h2, Depth: 5, AI Wins: 97, Random Wins: 3, Draws: 0, Avg Time: 17.81 seconds, Accuracy: 97.00%
=====
Heuristic: h3, Depth: 1, AI Wins: 75, Random Wins: 21, Draws: 4, Avg Time: 0.03 seconds, Accuracy: 75.00%
Heuristic: h3, Depth: 2, AI Wins: 69, Random Wins: 29, Draws: 2, Avg Time: 0.14 seconds, Accuracy: 69.00%
Heuristic: h3, Depth: 3, AI Wins: 71, Random Wins: 26, Draws: 3, Avg Time: 1.58 seconds, Accuracy: 71.00%
Heuristic: h3, Depth: 4, AI Wins: 86, Random Wins: 13, Draws: 1, Avg Time: 11.28 seconds, Accuracy: 86.00%
Heuristic: h3, Depth: 5, AI Wins: 91, Random Wins: 8, Draws: 1, Avg Time: 125.32 seconds, Accuracy: 91.00%
=====

```

We played 100 games for each depth and each heuristics. Average time for each game written in results as “Avg. Time”.

Hardware Used

- **Processor:** AMD Ryzen 4800H
- **RAM:** 16 GB DDR4