

CSE4082 Project #2 - Othello

(Due 12.01.2025 at 23:59, electronic submission only, to cse.cse482@gmail.com)

In this project, you will implement the Othello (Reversi) game along with an AI player for this game.

Othello, also known as Reversi, is a strategic board game played between two players on an 8x8 grid. Each player has discs colored black on one side and white on the other. The goal of the game is to have the majority of discs showing your color by the end. Players take turns placing their discs on the board in a way that sandwiches one or more of their opponent's discs between two of their own, either horizontally, vertically, or diagonally. When this happens, the sandwiched discs are flipped to the player's color. Strategy revolves around controlling key positions, particularly the corners, as they are difficult for the opponent to flip. The game ends when neither player can make a valid move, and the winner is the one with the most discs of their color on the board. The rules of the game are explained in [1].

Initial Board Position:

The initial setup of the board is as follows (O represents white discs, and X represents black discs):

1
2
3
4	.	.	.	O	X	.	.	.
5	.	.	.	X	O	.	.	.
6
7
8
	a	b	c	d	e	f	g	h

Black player starts first.

Requirements:

You must implement a fully functional game (text-based interface is acceptable) that supports the following modes:

1. Human Player vs Human Player.
2. Human Player vs AI Player.
3. AI Player vs AI Player.

For the AI player, you are required to implement **minimax** with **alpha-beta pruning** (or **negamax with alpha-beta pruning**). Additionally, the AI should support three evaluation (heuristic) methods:

1. **h1** = (The number of player's discs placed – The number of opponent's discs placed).
2. **h2** = (To be designed by your group. For example, consider positional advantages such as controlling corners or edges).

3. **h3** = (To be designed by your group. For instance, consider mobility, i.e., the number of valid moves available).

The AI player's complexity should be configurable by:

- The number of plies (depth of the tree).
- The evaluation heuristic to be used (h1, h2, or h3).

Project Deliverables:

1. **Source Code:**

- Submit only the source code (no executable files).
- Include a README file explaining how to run your program.

2. **Design Document:**

- Describe the classes, their fields, and methods used in your project.
- Provide a detailed explanation of the evaluation methods (h1, h2, h3).
- Include the maximum ply number that can be achieved along with the maximum time required to find the best move. Specify the hardware used for this measurement.

3. **Videos:**

- Record the following gameplay scenarios (compress videos if necessary):
 1. AI player using **h1** vs AI player using **h2** (using the maximum ply possible for each player).
 2. AI player using **h2** vs AI player using **h3** (using the maximum ply possible for each player).
 3. AI player using **h1** vs AI player using **h3** (using the maximum ply possible for each player).
 4. Human Player vs the Best AI Player Configuration.
- Ensure the videos clearly show gameplay and the AI's decision-making process. Videos that are unclear or missing will result in your project being considered non-functional.

Additional Notes:

- The project must be done in groups of three.
- You are not allowed to use any AI tools or pre-existing source code or libraries.

References:

[1] Wikipedia, Reversi, <https://en.m.wikipedia.org/wiki/Reversi>. Accessed Dec 2024.