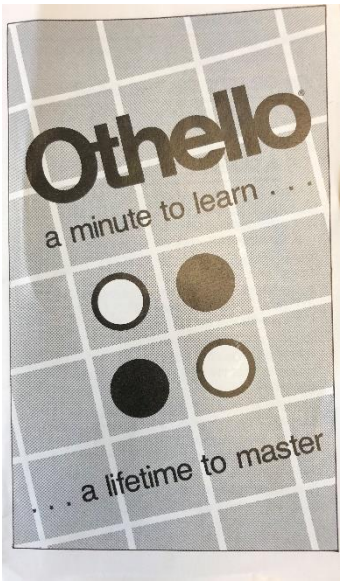
Computer Science 4753 – Fall 2025 – Assignment 3
**An Intelligent Othello Player**
Due: Monday, November 10, 2025 (by 7am)




This assignment will give you experience implementing a classical two player game, Othello, along with an AI opponent driven by the mini-max heuristic search algorithm. (See below for game rules.)

You have two weeks (14 days) to accomplish this assignment. I strongly advise you to begin **immediately** to ensure you are able to complete the assignment on time – especially as this program will be due Monday of finals week and you will need to reserve time to study for your various finals.

As you begin to think about the assignment I'd encourage you to break it down into two relatively separate tasks:

(1) implementing the mechanics of the game Othello; inclusive of representing the board, initializing play, accepting player moves (including rejecting illegal moves), computing the discs that should be flipped, displaying the board after each move (ASCII graphics are acceptable), score keeping, and recognition of when the game is complete.
(2) Implementing the Mini-Max algorithm, with alpha-beta pruning, and a reasonable heuristic.

You may choose your implementation language. Python would certainly be a reasonable choice.

Requirements (including Grading Rubric: [110/100])

1. [20%] Your program is appropriately commented, including a comment block at the beginning of your main program with: your name, your student number, the date, the assignment number, and a brief description of what the program does [10%]. Comments should also be present throughout the program to explain what each part does [10%].

2. [40%] Your program implements the rules of the Othello game accurately. In other words, two humans can use your program to successfully play Othello. Details provided in Task (1) above.

3. [20%] Your program properly implements the Mini-Max algorithm. Search depth should be easily adjustable and the program should display the total number of game states examined prior to making a move. Your program should support computer playing either white or black. Note: to demonstrate that mini-max is implemented properly, you should have a debug mode you can switch on (on a move by move basis) that shows all of the sequences of moves considered from the current state and the heuristic value associated with each move sequence. This debug mode should be OFF by default. PLEASE NOTE: YOU WILL NOT BE ABLE TO DEMONSTRATE MINI-MAX IF YOU HAVE NOT FULLY IMPLEMENTED THE RULES FOR OTHELLO FIRST.

4. [20%] Your program properly implements alpha-beta pruning. You will provide the ability to easily switch on or off alpha-beta pruning on a move by move basis. By observing the total number of game states when alpha-beta is enabled verses disabled, you should be able to demonstrate that alpha-beta pruning is operating properly.

5. [10%] Awarded if your program is able to beat a human (who is really trying to win). This will be demonstrated by a recorded program trace of an entire game.

6. You must work independently and develop your own code. Sharing of code, or use of an LLM to generate code, is ABSOLUTELY forbidden. You may, of course, look at solutions to this problem that are online, but plagiarizing a solution from the internet is **strictly forbidden**. If you attempt to do so, it will be obvious as you must be able to explain how your code operates during a one-on-one code review session with the instructor, and if you don't write the program yourself you won't be able to do so.

**Submitting your assignment for grading:**

You will need to submit TWO separate documents: (a) a copy of your source code, and (b) an execution trace of your program playing you in a game of Othello from first more to last. A copy of these two documents, combined into a single ZIP file, should be emailed to me (mike@LaTech.edu) **no later than 7am on Monday, November 10th**. The subject line of the email should be: *CSC 475 : Othello Program : <your name>* where *<your name>* is replaced with your name. So, for example, if Isaac Asimov were enrolled in this class, his email would use the subject line: *CSC 475 : Othello Program : Isaac Asimov*

You will be required to meet with me for approximately 15 minutes on Monday, November 10th to demo your program and explain how it works. A sign-up sheet will be available online no later than Friday, November 7th. These meetings will determine your grade on the program. If you cannot explain to my satisfaction how your code works, or make simple changes to it, you will receive a grade of zero on the program (which being less than 50% will result in failure of the course). Now, don't panic! If you wrote your own program you will find your 15 minute meeting to be exceptionally easy to get through.

Be prepared to demo your program on your own laptop.

# Othello Rules

## CONTENTS

Othello gameboard with 64 discs, which are black on one side, white on the other.

## OBJECT OF THE GAME

The object of the game is to have the majority of your color discs on the board at the end of the game.

## A MINUTE TO LEARN

Open disc drawers. See Figure 1. Each player chooses one color disc to use throughout the game.

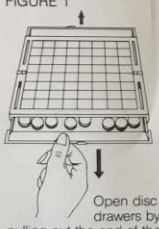
Black places two black discs and White places two white discs as shown in Figure 2. Game always begins with this set-up.

On your move, remove one disc at a time from the disc drawer and place it on the gameboard.

A move consists of "outflanking" your opponent's disc(s), then *flipping* the outflanked disc(s) to your color.

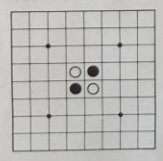
To outflank means to place a disc on the board so that your opponent's row (or rows) of discs is bordered

FIGURE 1



Open disc drawers by pulling out the end of the gameboard, as shown.

FIGURE 2





at each end by a disc of your color. (A "row" may be made up of one or more discs.)

Here's one example: White disc A was already in place on the board. The placement of white disc B outflanks the row of three black discs.

Ⓐ ● ● ● Ⓑ

White flips the outflanked discs and the row now looks like this:

Ⓐ ⊙ ⊙ ⊙ Ⓑ

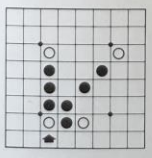
## OTHELLO® RULES

1. Black always moves first.

2. If on your turn you cannot outflank and flip at least one opposing disc, your turn is forfeited and your opponent moves again. However, if a move is available to you, you may not forfeit your turn.

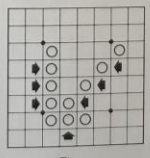
3. A disc may outflank any number of discs in one or more rows in any number of directions at the same time—horizontally, vertically or diagonally. (A row is defined as one or more discs *in a continuous straight line.*). See Figures 3 and 4.

FIGURE 3



Disc placed here

FIGURE 4



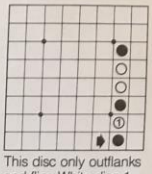
These discs flipped



4. You may not skip over your own color disc to outflank an opposing disc. (See Figure 5.)

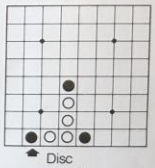
5. Disc(s) may only be outflanked as a direct result of a move and must fall in the direct line of the disc placed down. See Figures 6 and 7.

FIGURE 5



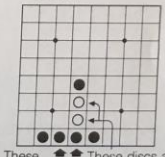
This disc only outflanks and flips White disc 1.

FIGURE 6



Disc placed here

FIGURE 7



These discs flipped

These discs are not flipped (even though they appear to be outflanked)

6. All discs outflanked in any one move must be flipped, even if it is to the player's advantage not to flip them at all.

7. A player who flips a disc which should not have been turned may correct the mistake as long as the opponent has not made a subsequent move. If the opponent has already moved, it is too late to change and the disc(s) remain as is.

8. Once a disc is placed on a square, it can never be moved to another square later in the game.

9. If a player runs out of discs, but still has an op-



portunity to outflank an opposing disc on his or her turn, the opponent must give the player a disc to use. (This can happen as many times as the player needs and can use a disc.)

10. When it is no longer possible for either player to move, the game is over. Discs are counted and the player with the majority of his or her color discs on the board is the winner.
NOTE: It is possible for a game to end before all 64 squares are filled.

## SAMPLE GAME

Black moves first. A black disc can be placed on square C4, D3, E6 or F5 to outflank a white disc. Black decides to place a disc on square C4. (See Figure 8.)

The outflanked white disc between the two black discs is flipped over. (See Figure 9.)

FIGURE 8

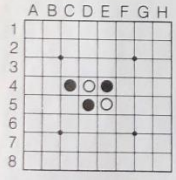
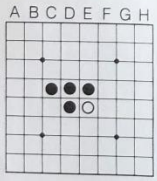



FIGURE 9




It is now White's turn. A white disc can be placed on square C3, E3 or C5 to outflank a black disc. White decides to place a disc on square C3. (See Figure 10).



The outflanked black disc between the two white discs is flipped over. (See Figure 11.)

FIGURE 10

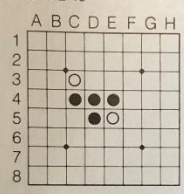
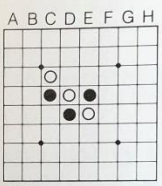



FIGURE 11




The game continues in this way until neither player can move. The player with the most discs at this point is the winner.

## SCORING

Players desiring to score their games may do so by determining the margin by which a player won a game. Simply subtract the smaller number of discs from the larger.

Players may also set up their own methods of scoring. For example, establish a predetermined number of games or points to win by in a series of games.

After the game, return discs to the disc drawers for storage and close up the gameboard.

©1986 by Milton Bradley Co. All Rights Reserved.

4713-X1