

COMP 4106 - ARTIFICIAL INTELLIGENCE
WINTER 2015

ASSIGNMENT #2

DUE DATE: MARCH 9, 2015

Game Playing with MiniMax

Introduction

In this assignment you will be implementing the game Othello, also called Reversi.

Othello

Othello is played on an 8-by-8 board, where players take turns placing discs, with one side being black, and the other being white. Play begins with each player having two pieces on the board, arranged as in Figure ?? . Typically, black plays first.

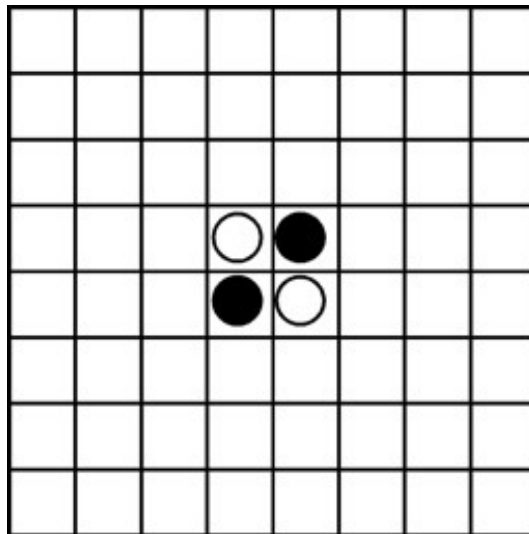


Figure 1: Starting position for Othello

During their turn, players may place a single piece on the board in any position where there exists a straight line (horizontal, vertical, or diagonal) between the new piece, and an existing piece controlled by the player, with any number of opponent-controlled pieces filling this line (at least one). When this piece is placed, all intermediate opponent pieces are flipped or replaced, and the player now controls them. It is possible for a player to “flip” more than one line in a single move.

If a player cannot make a valid move, play passes to the other player. The game ends when neither player can make a valid move, at which point the player who controls the most discs wins the game. A more detailed explanation of the rules may be found here: <http://en.wikipedia.org/wiki/Reversi#Rules>

Assignment Objectives

- Implement MiniMax search with Alpha-Beta pruning for Othello.
 - Implement two different heuristics for Othello
 - Enable player *vs* computer play of the game, where the computer player uses one of your heuristics. You are not being marked on the usability of your player interface, so long as it is easily readable.

- Enable computer *vs* computer play of the game, where each computer player uses a different heuristic.
- Provide a way to bound the depth of the search.
- Code your assignment in such a way as to be able to show every move being made in both of the games.
- Provide a way to measure and record the **Node Count** of the computer player's search. The Node Count is the number of nodes visited by the Mini-Max algorithm, excluding those pruned by alpha-beta pruning.
- Write a short report (no more than one (1) page) about the state space of the game, and about the choice of your heuristics, and the **Node Count** you had for the different options. *Please bring a hard copy of this report with your name and student number to your demo.*

Questions

During the demo you should be prepared to discuss the following questions:

- Explain the heuristics you used for each of the games.
- In each of the games, does one player always win?
- How does the Node Count change from Mini-Max without alpha-beta, to Mini-Max with alpha-beta enabled?

Tips

Don't spend too much time on the graphics. A command line representation is fine, so long as it is understandable.

Bonus

Implement Iterative Deepening, using the Principal Variation to achieve move ordering. You may also choose to implement a different move ordering algorithm such as Killer Moves or the History heuristic, but they are more complex. Be prepared to demonstrate how this impacts the Node Count of your search. (10% Bonus)