COMP 4106 - Artificial Intelligence
Winter 2018

Assignment #2

Due date: March 12, 2018

Game Playing with MiniMax

Introduction

In this assignment you will be implementing an obscure board game "Overboard".

Overboard

Overboard is played on a 6x6 board that begins completely covered with the players' pieces, for example in the pattern shown in Figure 1.

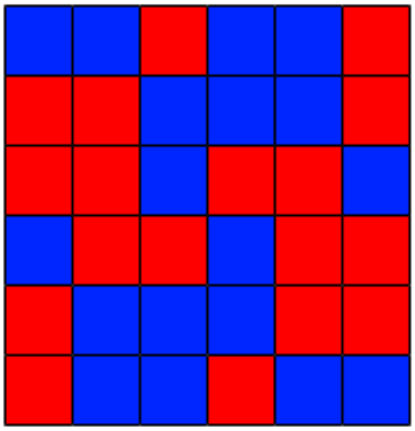

Figure 1: Starting position for Overboard

During their turn, a player may "push" one or more of the opponent's pieces "overboard" (off the boundary of the board) by moving one of their pieces in a straight line towards that boundary (imagine on a physical board the piece sliding the opponents' pieces off the board). So long as at least one piece is pushed "overboard", the player may move any number of squares towards the boundary, pushing any pieces they want, however they may not push their own pieces off the board. Diagonal moves are not possible.

If a player cannot or does not want to push the opponent's pieces off the board, they may instead move one piece vertically or horizontally one square to an empty space.

A possible move is shown in Figure 2. Here, the red player has moved the highlighted (orange) piece two squares up, pushing two of blue's pieces off the board.

Remember to consider all possible moves that can be made from each piece. For example, it would be possible in the above image for Red to move his piece in (1, 3) (third from the left on the top) down two squares, pushing the entire column of pieces (including his own) down two and pushing two of Blue's pieces off the board. Also, remember that the move is legal so long as it is in a straight line, pushes at least one of the opponent's pieces off, and does not push any of his own pieces off the board.

The play continues until one player has lost all of his pieces, at which point he is eliminated and the other player wins.

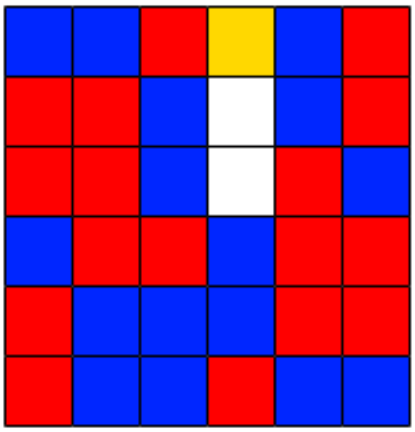

Figure 2: A Possible Move for Red in Overboard (the orange piece was moved)

Assignment Objectives

- Implement MiniMax search with Alpha-Beta pruning for Overboard.
  - Implement a heuristic-based solution.
  - Enable a computer *vs* computer play of the game, where one player uses the heuristic and the other plays random moves.
  - Implement two different heuristics for the game.
  - Enable a 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.
- 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 MiniMax algorithm, excluding those pruned by Alpha-Beta pruning.
- Code your assignment in such a way so as to be able to show every move being made in both of the games.
- Write a short report (no more than 2 (two) pages) about the state space you have used, about the choice of your heuristics, and the Node Count you had for the different options, and why they have worked the way they did.

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 MiniMax without Alpha-Beta, to MiniMax with Alpha-Beta enabled?

Tips

Don't spend too much time on the graphics. a command line based UI is sufficient. Also, I suggest that you implement a working and tested implementation of MiniMax first, and that you then use it for Overboard.

Bonus

Can you think of a way to improve the number of nodes pruned with Alpha-Beta pruning in Overboard?