

COMP 4106 - ARTIFICIAL INTELLIGENCE  
WINTER 2017

ASSIGNMENT #2

DUE DATE: MARCH 9, 2017

# Game Playing with MiniMax

## Introduction

In this assignment you will be implementing the game “Focus”, and using AI techniques to play it as the perspective player.

## Focus

Focus, also called Domination, is a commercial board game first published in 1964. It is played on an unorthodox 8-by-8 board, where the three squares closest to each corner are removed from the board. Unlike other games of this type, pieces may be “stacked” on top of one another, to a maximum height of five pieces. During a player’s turn, he or she may move a stack he or she “controls” (indicated by their piece being the top piece) any number of squares, vertically or horizontally, equal to or less than the height of the stack. Players may also “split” a stack, moving the top three pieces of a five piece stack, for instance, up to three squares in any direction. When a stack is moved into a square occupied by another stack, they are merged, with the first stack (the one that was moved) placed on top of the second. If the resultant stack is larger than five pieces, all those pieces below the first five are “captured” by the player who moved the stack.

A graphical illustration of the game’s *symmetric* starting position, along with its board structure, is shown in Figure 1. You must implement your game with the positions of the pieces being placed randomly.

The game starts with each player having 18 pieces. In the initial configuration, whether the board is symmetric or random, the outer ring of the board (top and bottom rows, leftmost and rightmost columns) is not used at all. However, after the game begins, players are free to move stacks into the outer ring of originally unoccupied squares.

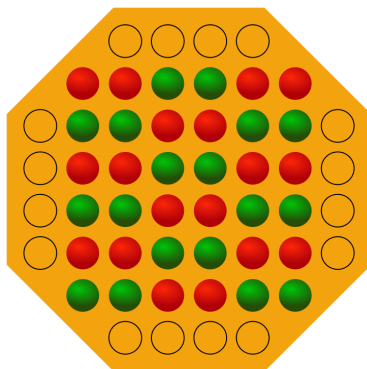


Figure 1: The *symmetric* starting position of the game Focus.

In the formal rules of the game, play proceeds until one player cannot make a valid move (they control no stacks and have no reserve pieces), in which case their opponent has won. However, this may take a long time. Alternatively, you can set your win condition to be that the player has captured a certain number of their opponent’s pieces (say, 8) *or* the opponent has no valid moves. Please be prepared to discuss your win condition during your demo!

As Focus with its *symmetric* starting position has the same starting position every time, you may see a tendency for identical play to arise. A common method to avoid this is to add a small, random factor to your heuristics (not large enough to change its ranking on different boards) that can break ties. But you should be implementing it with the general *randomized* starting configuration.

## Assignment Objectives

- Implement MiniMax search with Alpha-Beta pruning for Focus.
  - Implement two different heuristics for Focus.
  - 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.
- Code your assignment in such a way that you are able to show every move being made in your game.

## Questions

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

- Explain the heuristics you used for the game.
- Does one player always win in the game?

## Tips

Don't spend too much time on the graphics. Command line based I/O is fine.

## Bonus

1. Implement Focus with the self-capture rule: If a player captures his or her own piece, it is placed in “reserve” and, in lieu of moving a stack, he or she may place that piece back on the board in *any* valid square, including on top of a stack. Please be aware that it changes the strategies of the game significantly, as capturing your own piece changes from a desirable to an undesirable action.
2. Implement Focus for *four* players. You may use any of the multi-player game playing techniques discussed in class. The starting position for four-player Focus is shown below.

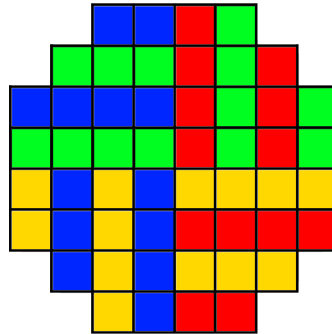


Figure 2: The starting position of the game Focus for four players