Connect Four

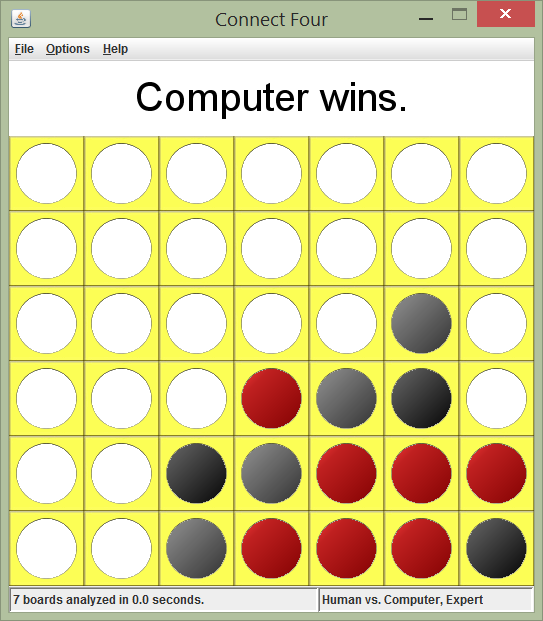
# Overview

In this series of lab exercises, we will explore using 2-D arrays to create a playable version of the game Connect Four.

## Objectives

* Understand how to represent and implement the game board in code.
* Demonstrate how to traverse arrays to evaluate if a player has won the game

# Exercise 1 – The Game

[Connect Four](http://en.wikipedia.org/wiki/Connect_Four)[[1]](#footnote-1) is a two-player game in which each player chooses a color and then drops checkers of that color into a grid. The checkers fall straight down and come to rest in the first available cell within the column. The object of the game is to be the first player to position four checkers in a row, column, or diagonal.  
  
This Java applet offers the traditional two-player mode as well as the option to play against the computer. The player can choose whether or not to go first. The game has an AI engine based on the [minimax algorithm](http://en.wikipedia.org/wiki/Minimax)[[2]](#footnote-2) with [alpha-beta pruning](http://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning)[[3]](#footnote-3). Strictly speaking, the computer first looks for a winning move. If no winning move is found, it blocks the opponent's winning move, if one exists. If the opponent has no winning move, it runs the minimax algorithm up to the designated depth. Beginner, intermediate, advanced, and expert levels correspond to a depth-first search of 2, 4, 6, and 8 levels, respectively.  
  
Overall, the AI engine is a strong (though not perfect) opponent when searching 8 levels ahead. It analyzes several hundred thousand board configurations in a fraction of a second, so you're not waiting around for the computer to make its move. Download Exercise-1 from GitHub, <https://github.com/viperguynaz/ConnectFour-Exercise1> and run ConnectFour.jar (you should just have to double-click the file). Explore the game, then answer the following questions. Have fun playing!

Questions:

1. How can you represent the Connect Four game board in Java?
2. How can you represent each Connect Four player in Java?
3. How can you determine when a user selects a column to drop their checker, if that is a valid move?
4. After a player has dropped a checker, how can you determine if the player has won the game?
5. Given your representation of the Connect Four game board, write a snippet of code (not a complete method) to traverse each row.
6. Given your representation of the Connect Four game board, write a snippet of code (not a complete method) to traverse each column.
7. Given your representation of the Connect Four game board, write a snippet of code (not a complete method) to traverse each diagonal (major and minor).

1. http://en.wikipedia.org/wiki/Connect\_Four [↑](#footnote-ref-1)
2. http://en.wikipedia.org/wiki/Minimax [↑](#footnote-ref-2)
3. http://en.wikipedia.org/wiki/Alpha%E2%80%93beta\_pruning [↑](#footnote-ref-3)