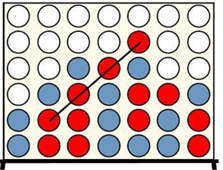
**Homework #5**

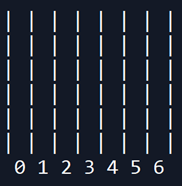
In this exercise, you will write a Monte Carlo RL agent to play Connect Four.

(Taken partially from: <https://en.wikipedia.org/wiki/Connect_Four>)

**Connect Four** is a game in which the players choose a color and then take turns dropping colored tokens into a six-row, seven-column vertically suspended grid. The pieces fall straight down, occupying the lowest available space within the column. The objective of the game is to be the first to form a horizontal, vertical, or diagonal line of four of one's own tokens. Connect Four is a [solved game](https://en.wikipedia.org/wiki/Solved_game) as the first player can always win by playing the right moves.

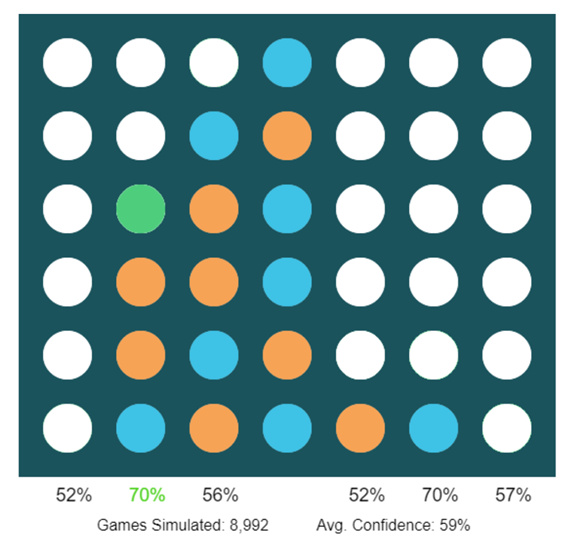
An example of a victory for a red player can be seen in the following figure:

In this homework assignment, I provide files play.py and game.py which have the basics of what is needed to play and the game and a skeleton for the MC agent for something that looks like this:



Your assignment: Implement the function inputMC so it plays an MC strategy.

As mentioned in class, this agent should make a move and then play a large number of games (100 is preferable, but 30 is fine if you computer is slow) to determine which column to choose. The agent then compiles the win/loss probabilities to choose the action with the highest value. This picture from the slide helps illustrate:



In this example, the agent will choose either the second or sixth column as each has the equal, but highest, probability of winning the game.

There are two ways I will check your agent:

1. Against the stupid agent, inputRandom. Your agent should beat this agent 100% of the time.
2. Against the simple heuristic agent, inputHeuristic. This agent isn’t bad, but your MC agent should still beat it nearly 100% of the time.

In theory you could check it by manually playing against the MC agent, but I will not check things this way.

Grade Breakdown:

60 points for correctly implementing the inputMC function

20 points for documenting your agent

20 for beating both agents nearly 100% of the time. Please include in your output a comparison of the win rates against both of these agents. This output could be either a printout in python (recommended) or a separate file.

Due date: December 11th 11:59 PM.

Submission by Github