Search Programming Assignment
Individual Assignment
Due: Feb 6th 11:59pm

For this assignment you will be developing an AI character to play Tic-tac-toe against.

What is Tic-tac-toe:
Tic-tac-toe is a classic two-player turn-based game in which players try to get three Xs or Os in a row on a 3x3 board.

Description:
You can use any algorithm we talked about in class: uninformed, informed, local, adversarial, or create your own. You will need to document your design choice in the supplemented documentation. The starting player should be selected randomly. Once the game starts, each player will take their turn. When it comes to the AI's turn, you will calculate the move for the AI player using the algorithm you designed.

The graphical representation of this game is up to you, however, the game should have a graphical representation.

Code:
- Use Python as the programming language for this assignment. *If you choose to use a programming language other than Python, please email the TA for approval.*
- For the assignment, you can either use GUI or text based interface for the assignment.
    - For text based interface. The user can for example input the row and column of where he/she wants to add their X/O move.
    - For the GUI based interface. The user can enter their move through mouse click position.

What to submit:
- Code
- Readme file: includes instructions to run your game
- Documentation: a 2-page document outlining the algorithm you chose to implement for your player implementation and why you made this choice. If you developed your own algorithm, please outline it here. Also describe how general your algorithm is and what type of environments is it suited for.

Where to submit your assignment:
- Please submit your deliverables into your SVN repository inside a folder name 'assgn1'.
- The path for the SVN repo is –
    - https://trac.ccs.neu.edu/svn/aispring2013/bbb_aaa
    - Replace 'bbb' with the first 3 characters of your last name

- Replace ‘aaa’ with the first 3 characters of your first name (use your middle name here if you have one)

Grading Criteria:

- Correctness (90 points)
- Readability/modularity (10 points)

If the computer player cannot be beaten, you will get full credit for the correctness criteria. For readability and modularity of your code you will receive 10% (to get full credit, add documentation for your code, incorporate any comments on your previous assignments, make sure you do not use random variable names, etc., *see the coding review document for more information*).