Assignment 5 – Tic Tac Toe

Goals

- Create a two player tic tac toe game
- Continue working with loops and lists
- Familiarity with functions (inputs, return values, no return values, using a main function)

Requirements

- You will create a program that uses functions to simulate a two player game of Tic Tac Toe. The program will allow the two players to place an "x" or an "o" somewhere on the board and determine when someone wins or when a stalemate is reached.
- In order to keep track of the tic tac toe board, you will use a list of 9 single character strings, each representing a spot on the board. The board begins with numbers (0-8):

```
0  1  2
3  4  5
6  7  8
```

  Which is represented in list form as:

```
["0", "1", "2", "3", "4", "5", "6", "7", "8"]
```

  Where the numbers on the board and in the list correspond with same index positions in the list. As the game advances, the numbers will be replaced with x's and o's.
- You may **not** use any global variables or constants in your code.

Functions provided for you

- To help you out, we have provided two functions for you to use: a tic tac toe solver (which tells the current state of the game) and a print board function (which displays the tic tac toe board)
  - In order to use these functions, you will need to place the given file, TicTacToeSolver.py, in the same directory as your assignment 5 file. To use a function in your code, place the line "`import TicTacToeHelper`" at the top of your code. Importing TicTacToeHelper gives you access to functions that were defined in a different file.
  - To use the solver function, the syntax is:
    `TicTacToeHelper.checkForWinner(boardList, move_counter)`
    - Inputs: a list representing the tic tac toe board, and an integer representing the total number of moves that have been made
    - Return value: a single character string:

- 'x' if x won
- 'o' if o won
- 'n' if there is not winner
- 's' if the game reached a stalemate (full board with no winners)

- To use the print board function, the syntax is:

```
TicTacToeHelper.printUglyBoard(boardList)
```

  - Input: a list representing the board
  - Return value: none
  - Takes the list representing the board and prints it out for the user to see (the board does not have to exactly match the sample outputs, but it should show the board as three rows)

Functions for you to create

- You will use the given function to create the game in addition to implementing the following functions:
  - **`isValidMove(boardList, spot)`**
    - Input: a list representing the board, an integer corresponding to the index position that a user would like to place their letter on
    - Return value: a boolean value
    - This function should look at the specified spot on the board and return **`True`** if the spot is open or **`False`** if the spot is taken or out of range
  - **`updateBoard(boardList, spot, playerLetter)`**
    - Input: a list representing the board, an integer corresponding to the index position that a user would like to place their letter on, a string representing the user's letter ("x" or "o")
    - Return value: none
    - Takes the current board list and places the player's letter in the specified spot on the board.
  - **`playGame()`**
    - Input: none
    - Return value: none
    - Initialize the list representing the board to a list of strings with the numbers 0-8.
    - Keep track of the total number of moves that have been made using a counter.
    - Using a while loop, allow each player to take a turn until the game ends. Note: only one player should take a turn per loop iteration, meaning you should use a variable that alternates between player x and player o.

        - At each turn, ask the current player what position they would like to put their letter on the board. Check that the selected spot is valid using **`isValidMove`** and then update the board with the move using **`updateBoard`** if the move is valid. If the spot is not valid, keep asking the user for a position on the board until their move is valid.
        - Use the **`checkForWinner`** method provided to you to control the while loop to determine when the game is over. Hint: the game is over only when either x or o wins, or if there is a stalemate.
        - Print out final board at the end of the game and a message about who won or if there was a stalemate.
    - **`main()`**
        - Input: none
        - Return value: none
        - Use a while loop to allow the user to keep playing a new game of tic tac toe until they decide to stop playing again.

Extra Credit

- Print pretty board:
    - Instead of printing the board out as shown above, define a new function called **`printPrettyBoard`** with the same inputs (board list) and return value (none) as the **`printUglyBoard`** function provided to you, but instead prints the tic tac toe board out to look like:

```
0 | 1 | 2
---------
3 | 4 | 5
---------
6 | 7 | 8
```

    - This function should use loops to print the board out. Wherever you print the board out in your original code, use this new function instead.
- Play against the computer:
    - Allow the user the option to play against the computer or another user. When playing against the computer, the computer's moves can be random.
- Allow the user to choose which player begins:
    - Give the user the option of starting the game as either player x or player o. This will determine who makes the first move.

Sample Output 1

```
Welcome to Tic Tac Toe!

0  1  2
3  4  5
6  7  8
Player x, pick a spot: -1
Invalid move, please try again.
Player x, pick a spot: 0

x  1  2
3  4  5
6  7  8
Player o, pick a spot: 0
Invalid move, please try again.
Player o, pick a spot: 4

x  1  2
3  o  5
6  7  8
Player x, pick a spot: 1

x  x  2
3  o  5
6  7  8
Player o, pick a spot: 1
Invalid move, please try again.
Player o, pick a spot: 2

x  x  o
3  o  5
6  7  8
Player x, pick a spot: 3

x  x  o
x  o  5
6  7  8
Player o, pick a spot: 6
```

```
x  x  o
x  o  5
o  7  8

Game Over!
Player o is the winner!
Would you like to play another round? (y/n): n
Goodbye!
```

Sample Output 2

```
Welcome to Tic Tac Toe!

0  1  2
3  4  5
6  7  8
Player x, pick a spot: 0

x  1  2
3  4  5
6  7  8
Player o, pick a spot: 1

x  o  2
3  4  5
6  7  8
Player x, pick a spot: 8

x  o  2
3  4  5
6  7  x
Player o, pick a spot: 4

x  o  2
3  o  5
6  7  x
Player x, pick a spot: 7
```

```
x  o  2
3  o  5
6  x  x
Player o, pick a spot: 6

x  o  2
3  o  5
o  x  x
Player x, pick a spot: 2

x  o  x
3  o  5
o  x  x
Player o, pick a spot: 5

x  o  x
3  o  o
o  x  x
Player x, pick a spot: 3

x  o  x
x  o  o
o  x  x

Game Over!
Stalemate reached!
Would you like to play another round? (y/n): y

0  1  2
3  4  5
6  7  8
Player x, pick a spot:.
.
.
.
(continue the game until the user does not want to play another round)
```

Deliverables and Submission Instructions

- Create a folder on your computer called
  `ITP115_a#_lastname_firstname`
  (replace # with this lab number)
- Inside the folder, include your python source code
- Compress the folder (make a zip file) called
  `ITP115_a#_lastname_firstname.zip`
  (replace # with this assignment number)
- Upload zip file to Blackboard site for our course

Grading

Item	Points
Tic Tac Toe	35
Total*	**35**

** Points will be deducted for poor code style, or improper submission.*