Assignment 8 – Functions 2

Learning Objective

Define and call functions to simulate a game of Tic Tac Toe.

Assignment Description

Write 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, create a list of strings where each string represents a place on the board. The list starts with position numbers and will be replaced with x's and o's. Start the list with the following values: "0", "1", "2", "3", "4", "5", "6", "7", "8". When the board is printed, the following is displayed:

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

As the user plays the game, the numbers will be replaced with x's and o's. Here's an example of what it may look like:

```
x  1  2
o  o  5
x  7  8
```

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). They are located in the TicTacToeHelper.py file provided.

- Put the TicTacToeHelper.py in the same folder as the Python file for this assignment.
- Place the following line at the top of your code below your top comment block: **`import TicTacToeHelper`**
- To use the solver function, the syntax is: **`TicTacToeHelper.checkForWinner(boardList, moveCounter)`**
    - Parameter 1: boardList is a list of strings representing the Tic Tac Toe board
    - Parameter 2: moveCounter is an integer representing the total number of moves that have been made
    - Return value: a single character string which is one of the following:

  - "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)`
  - Parameter: boardList is a list of strings representing the Tic Tac Toe 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).

Steps

1. In PyCharm (Community Edition), open an existing project (such as ITP115) or create a new project.
   - If you open an existing project, then create a new directory (probably under the Assignments directory) named **a8_*last_first*** where *last* is your last/family name and *first* is your preferred first name. Use lowercase letters.
   - If you create a new project, then name it **a8_*last_first*** where *last* is your last/family name and *first* is your preferred first name. Use lowercase letters.
2. In the project or directory, create a new Python file called **assignment8.py**. At the top of the file, put comments in the following format and replace the name, email, and section with your actual information:

```
# Name, USC email
# ITP 115, Spring 2022
# Section: number or nickname
# Assignment 8
# Description:
# Describe what this program does.
```

3. Put the **TicTacToeHelper.py** file in your a8_*last_first* directory.
   - Download the file from Blackboard under the item for this assignment.
   - Add it to the correct directory by dragging to a8_*last_first* directory in PyCharm.
4. Place the following line at the top of your code below your top comment block:

```
import TicTacToeHelper
```

5. Define the **isValidNumber(boardList, position)** function.
    - Parameter 1: a list representing the board
    - Parameter 2: an integer corresponding to the index position that a user would like to place their letter on
    - Return value: a boolean value (True or False)
    - This function should check the position variable to see if it is a valid number (between 0 and 8 inclusive) and then see if that position is available. Return True if the position is available. Return False if the position is not a valid index or if the position is taken. A position is taken if it equals "x" or "o".
6. Define the **updateBoard(boardList, position, playerLetter)** function.
    - Parameter 1: a list representing the board
    - Parameter 2: an integer corresponding to the index position that a user would like to place their letter on
    - Parameter 3: 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 position on the board.
7. Define the **playGame()** function.
    - Parameter: None
    - Return value: None
    - Create a variable to represent the board that is a list of strings with the following values: "0", "1", "2", "3", "4", "5", "6", "7", "8".
    - Create an int variable to keep track of the total number of moves that have been made. It is the move counter for the game.
    - Create a string variable that will be used to see if there is a winner. Set it to "n".
    - Using a while loop, allow each player to take a turn until the game ends. The game is over when the checkForWinner() function does not return an "n". Only one player should take a turn per loop iteration. To accomplish this you can create a variable before the while loop that will alternate between player "x" and player "o". Alternatively, you can use the modulus (%) operator on the move counter. Start with player "x".

- In the while loop, print the board by calling the **printUglyBoard()** function in the TicTacToeHelper. Use branching to determine the current player ("x" or "o").
- Depending on the who's turn it is, ask the user what position they would like to put their letter on the board. The grader will only enter integers.

```
Player x, enter a number: 0
Player o, enter a number: 8
```

- Use looping to make sure that the user enters a valid integer by calling the isValidNumber() function. Once you have a valid number, update the board with the move by calling the updateBoard() function.

```
Player x, enter a number: -1
Player x, enter a number: 9
Player x, enter a number: 8
```

- After the branching, call the checkForWinner() function, which is located in the TicTacToeHelper module, to determine if the game is over. The game is over when the checkForWinner() function does not return an "n". Make sure to set the return value to the variable you created to control the while loop.
- After the while loop, print out the board and print the following message:

```
Game Over!
```

- Print out the correct statement based on the return value from checkForWinner(). You should not call the function again; use the correct variable.

```
Stalemate reached.
Player x is the winner!
Player o is the winner!
```

8. Define and call the **main()** function.
   - Parameter: None
   - Return value: None
   - Print the following message:

   ```
   Let's play Tic Tac Toe!
   ```

   - Call the playGame() function to play a round of Tic Tac Toe.

- Use a while loop to allow the user to keep playing a new game of Tic Tac Toe if they enter "y" or "Y".

```
Would you like to play another round (y or n)? y
```

9. Be sure to comment your code. This means that there should be comments throughout your code. Put a comment block before each function stating the parameters, return values, and what that function does. Points will be deducted for not having comments.
10. Follow coding conventions. You should use lowerCamelCase or snake_case for variable names. You are welcome to create any variables that you need.
11. Test the program. Look at the Sample Output below. Assignments that do not run are subject to 20% penalty.
12. Prepare your submission:
    - Find the **a8_*last_first*** folder on your computer and compress it. This cannot be done within PyCharm.
    - On Windows, use **File Explorer** to select the folder. Right click and select the Send to -> Compressed (zipped) folder option. This will create a zip file.
    - On Mac OS, use **Finder** to select the folder. Right click and select the Compress "*FolderName*" option. This will create a zip file.
13. Upload the zip file to your Blackboard section:
    - On Blackboard, navigate to the appropriate item.
    - Click on the specific item for this assignment.
    - Click on the **Browse Local Files** button and select the zip file.
    - Click the **Submit** button.

Grading

- This assignment is worth 35 points.
- Make sure that you the program runs. Points will be taken off if the graders have to edit the source code to test your program.
- Make sure to submit your assignment correctly as described above. Points will be taken off for improper submission.

Item	Points
TicTacToeHelper in directory	2
isValidNumber()	5
updateBoard()	3
playGame()	20
main()	5
Total	**35**

Sample Output

```
Let’s play Tic Tac Toe!

0  1  2
3  4  5
6  7  8
Player x, enter a number: -1
Player x, enter a number: 9
Player x, enter a number: 0

x  1  2
3  4  5
6  7  8
Player o, enter a number: 0
Player o, enter a number: 4

x  1  2
3  o  5
6  7  8
Player x, enter a number: 1

x  x  2
3  o  5
6  7  8
Player o, enter a number: 1
Player o, enter a number: 2

x  x  o
3  o  5
6  7  8
Player x, enter a number: 6

x  x  o
3  o  5
x  7  8
Player o, enter a number: 3

x  x  o
o  o  5
x  7  8
Player x, enter a number: 5
```

```
x  x  o
o  o  x
x  7  8
Player o, enter a number: 7

x  x  o
o  o  x
x  o  8
Player x, enter a number: 8

x  x  o
o  o  x
x  o  x

Game Over!
Stalemate reached.
Would you like to play another round (y or n)? Y

0  1  2
3  4  5
6  7  8
Player x, enter a number: 4

0  1  2
3  x  5
6  7  8
Player o, enter a number: 2

0  1  o
3  x  5
6  7  8
Player x, enter a number: 8

0  1  o
3  x  5
6  7  x
Player o, enter a number: 0

o  1  o
3  x  5
6  7  x
Player x, enter a number: 1
```

```
o  x  o
3  x  5
6  7  x
Player o, enter a number: 3

o  x  o
o  x  5
6  7  x
Player x, enter a number: 7

o  x  o
o  x  5
6  x  x

Game Over!
Player x is the winner!
Would you like to play another round (y or n)? q
Goodbye!
```

Extra Credit (options)

- 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.