

CMPUT175-Lab 3 Exercises

The solutions to these exercises have to be shown to your TA in your assigned Lab before the end of your lab time.

The exercises are about building the game of Tic Tac Toe. Tic Tac Toe is a game for two players who take turns to fill a 3 X 3 grid with either o or x. Each player alternates choosing an open space in the grid to mark with either x for a player or o for the other player. The goal (to win) is to have three xs or 3 os in a row, column or diagonal. The game can also end with a draw if the grid is full but no line is formed. The exercises combined will allow us to implementing Tic Tac Toe for two human players to play it on a computer.

1. Write a program that displays the Tic Tac Toe board. We have a python class in the provided file **Lab3Exercise.py** that initializes a board with a list of empty characters.

```
class TicTacToe:
    def __init__(self):
        # "board" is a list of 10 strings representing the board (ignore index 0)
        self.board = [" "]*10
        self.board[0]="#"
```

The constructor `__init__` creates a list of 10 one character strings. The first one at index 0 will be ignored (initialized with some character but not space) and the other 9 represent the 9 cells of the Tic Tac Toe grid. Write the method `assignMove(self, cell, ch)` that requests the Tic Tac Toe object to assign the symbol/letter `ch` to the position `cell` of the grid. The argument `ch` is assumed to be either "x" or "o". However, `cell` is a number and should be between 1 and 9. Write the method `drawBoard(self)` that would display the board as follows:

x x o	7 8 9
-----	-----
o o x	4 5 6
-----	-----
x o x	1 2 3

The first grid on the left represents the current state of the board while the grid on the right is static, and represents the index of each cell (like the numeric pad on the keyboard). The two grids are separated by a tab (\t).

Once you wrote your methods, create an object Tic Tac Toe with `myBoard=TicTacToe()` fill some cells in the board using for example `myBoard.assignMove(4,"x")` to put an x in the 4th cell, then display the board.

If the list board of the instance TicTacToe contains ["#", "x", "o", "x", "o", "o", "x", "x", "x", "o"], the board should look as above. If the list contains ["#", "o", "o", "o", "x", "x", "x", "x", "x", "o"] the board should look like:

x x o	7 8 9
-----	-----
x x x	4 5 6
-----	-----
o o o	1 2 3

If the list contains ["#", "o", "o", "o", "o", " ", "o", "o", "o", "o"] the board should look like:

o o o	7 8 9
-----	-----
o o	4 5 6
-----	-----
o o o	1 2 3

Write the methods **boardFull(self)** to check if the board is full, and **cellIsEmpty(self, cell)** to check whether a given cell is available. Make sure the argument cell is correct and will not generate an error of out of bound. If it is out of bound, the method should return **False**.

Write a small program to use and test these methods.

2. Continue the class TicTacToe by completing the method **isWinner(self, ch)** which returns True if the symbol in **ch** forms either a horizontal line, a vertical line or a diagonal line, and False otherwise. A line is formed by three consecutive cells with the same symbol (as in **ch**). This method is called within the method **whoWon(self)** provided in the python file.

Write a small program that uses and test these methods to fill the cells with **assignMove** check whether the board is full, whether a given cell is available, who won and display the board.

3. Write a program that allows two players, x and o, to alternate and provide their moves on the board. For instance asking the question: **It is the turn for x . What is your move?**

The program should check the input and only allow a valid cell between 1 and 9. Also, a cell that is not available (already filled) should not be allowed as a new move. The program should iterate until the game is over and display who won or whether there is a draw.

Here are some examples of output at the end of a game.

x x o	7 8 9
-----	-----
o o x	4 5 6
-----	-----
x o x	1 2 3

It's a tie.

x o o	7 8 9
-----	-----
x x o	4 5 6
-----	-----
x	1 2 3

x wins. Congrats!