Version_One

February 18, 2018

1 Tic Tac Toe

1.1 How to Play

There are two players for the game.

The goal of the game is to get three noughts or crosses aligned within a 3x3 grid.

This can occur horizontally, vertically or diagonally.

If there no noughts or crosses aligned, the players have drawed and the game starts again.

2 Guidance code

```
In [ ]: def horizontal(n): # function to make the horizontal lines of the grid
            for a in range(n):
                print(" ---", end="")


        def vertical(n): # function to make the vertical lines of the grid
            for a in range(n):
                print("|   ", end="")
            print("|", end="")


        def draw_board(n): # Makes board by combining the horiszontal and vertical functions
            for a in range(n):
                horizontal(n)
                print("")
                vertical(n)
                print("")
            horizontal(n)


        if __name__ == '__main__':
            board_size = int(input("Enter board size: "))
            draw_board(board_size)
```

3 My code

Unfinished due to messy coding use different technique

```
In [3]: import random
        def horizontal(n): # to print the horizontal lines of the board
            for horiz in range (3):
                print (" ---",end="")

        def vertical(n):# to print the vertical lines of the board
            for vert in range (3):
                print ("|   ", end=" ")
            print ("|", end="")

        def drawBoard(n):# to make compile the two function to create the board
            for board in range(n):
                horizontal(n)
                print("")
                vertical(n)
                print("")
            horizontal(n)

        drawBoard(3)

        board = [[0,0,0],[0,0,0],[0,0,0]] #board co-ordinates placeholde for the X and O
        positons = [0,1,2]

        def checkWin1():# Winning coordinate for the X and O in the diagonal direction to the ri
            if board [0][0] == "X" and board [1][1] == "X" and board [2][2] == "X":
                print("Player X Wins!!!")
            elif board [0][0] == "O" and board [1][1] == "O" and board [2][2] == "O":
                print ("Player O Wins!!!")

        def checkWin2(): # Winning coordinates for the X and O in the diagonal diraction to the
            if board [0][2]== "X" and board [1][1] == "X" and board [2][0] == "X":
                print ("Player X Win!!!")
            elif board [0][2]== "O" and board [1][1] == "O" and board [2][0] == "O":
                print ("Player O Win!!!")

        def checkWin3():# Winning coordinates for the X and O in the first horizontal
            if board [0][0] == "X" and board [0][1] =="X" and board [0][2] == "X":
                print ("Player X Win!!!")
            elif board [0][0] == "O" and board [0][1] == "O" and board [0][2] == "O":
                print ("Player O Win!!!")

        def checkWin3():# Winning coordinates for the X and O in the second horizontal
            if board [1][0]== "X" and board [1][1] == "X" and board [1][2] == "X":
                print ("Player X Win!!!")
```

```
    elif board [1][0]== "O" and board [1][1] == "O" and board [1][2] == "O":
        print ("Player O Win!!!")

def checkWin4():# Winning coordinates for the X and O in the third horizontal
    if board [2][0] == "X" and board [2][1] == "X" and board [2][2] == "X":
        print ("Player X Win!!!")
    elif board [2][0] == "O" and board [2][1] == "O" and board [2][2] == "O":
        print ("Player O Win!!!")

def checkWin5():# Winning coordinates for the X and O in the first Column
    if board [0][0] == "X" and board [1][0] == "X" and board [2][0] == "X":
        print ("Player X Win!!!")
    elif board [0][0] == "O" and board [1][0] == "O" and board [2][0] == "O":
        print ("Player O Win!!!")


def checkwin7():# Winning coordinates for the X and O in the second Column
    if board [0][1] == "X" and board [1][1] == "X" and board [2][1] == "X":
        print ("Player X Win!!!")
    elif board [0][1] == "O" and board [1][1] == "O" and board [2][1] == "O":
        print ("Player O Win!!!")

def checkwin8():# Winning coordinates for the X and O in the third Column
    if board [0][2] == "X" and board [1][2] == "X" and board [2][2] == "X":
        print ("Player X Win!!!")
    if board [0][2] == "X" and board [1][2] == "X" and board [2][2] == "X":
         print ("Player O Win!!!")

playerCharact = "X" or "O" or "x" or "o"

PlayerCharact = input ("Which character do you want nought(O) or crosses(X) ")
# userPlacement = raw_input("Where do you want to place your character")
user_x =(input("Enter the row to place character from 0-2 "))
user_y =(input("Enter the column to place chracter 0-2 "))

# print (board(str([user_x]))+(str([user_y]))+[charactSel])

# print (board[user_x][user_y])
```

```
 --- --- ---
|   |   |   |
 --- --- ---
|   |   |   |
 --- --- ---
|   |   |   |
 --- --- ---Which character do you want nought(O) or crosses(X) X
Enter the row to place character from 0-2 0
Enter the column to place chracter 0-2 2
```

```
    ---------------------------------------------------------------------------

    TypeError                                 Traceback (most recent call last)

    <ipython-input-3-1459d8ec05c9> in <module>()
     85     return (print (drawboard))
     86 
---> 87 boardPlacement()
     88 
     89 


    <ipython-input-3-1459d8ec05c9> in boardPlacement()
     80     matrix = []
     81     board = [[0,0,0],[0,0,0],[0,0,0]]#board co-ordinates placeholde for the X and O
---> 82     board.append(user_x)(user_y)
     83     matrix.append(board)(characterSel)
     84     drawboard.append(matrix)


    TypeError: 'NoneType' object is not callable
```