|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **21-05-2020** | **Name:** | **Varun G Shetty** |
| **Course:** | **Python** | **USN:** | **4AL17EC093** |
| **Topic:** | **Milestone project** | **Semester & Section:** | **6th ‘B’** |
| **GitHub Repository:** | **Varunshetty4** |  |  |

**Report :**

This solution follows the same basic format as the Complete Walkthrough Solution, but takes advantage of some of the more advanced statements we have learned. Feel free to download the notebook to understand how it works!

def display\_board(a,b):

print('Available TIC-TAC-TOE\n'+

' moves\n\n '+

a[7]+'|'+a[8]+'|'+a[9]+' '+b[7]+'|'+b[8]+'|'+b[9]+'\n '+

'----- -----\n '+

a[4]+'|'+a[5]+'|'+a[6]+' '+b[4]+'|'+b[5]+'|'+b[6]+'\n '+

'----- -----\n '+

a[1]+'|'+a[2]+'|'+a[3]+' '+b[1]+'|'+b[2]+'|'+b[3]+'\n')

display\_board(available,theBoard)

def display\_board(a,b):

print(f'Available TIC-TAC-TOE\n moves\n\n {a[7]}|{a[8]}|{a[9]} {b[7]}|{b[8]}|{b[9]}\n ----- -----\n {a[4]}|{a[5]}|{a[6]} {b[4]}|{b[5]}|{b[6]}\n ----- -----\n {a[1]}|{a[2]}|{a[3]} {b[1]}|{b[2]}|{b[3]}\n')

display\_board(available,theBoard)

def place\_marker(avail,board,marker,position):

board[position] = marker

avail[position] = ' '

def win\_check(board,mark):

return ((board[7] == board[8] == board[9] == mark) or # across the top

(board[4] == board[5] == board[6] == mark) or # across the middle

(board[1] == board[2] == board[3] == mark) or # across the bottom

(board[7] == board[4] == board[1] == mark) or # down the middle

(board[8] == board[5] == board[2] == mark) or # down the middle

(board[9] == board[6] == board[3] == mark) or # down the right side

(board[7] == board[5] == board[3] == mark) or # diagonal

(board[9] == board[5] == board[1] == mark)) # diagonal

def random\_player():

return random.choice((-1, 1))

def space\_check(board,position):

return board[position] == ' '

def full\_board\_check(board):

return ' ' not in board[1:]

def player\_choice(board,player):

position = 0

while position not in [1,2,3,4,5,6,7,8,9] or not space\_check(board, position):

try:

position = int(input('Player %s, choose your next position: (1-9) '%(player)))

except:

print("I'm sorry, please try again.")

return position

def replay():

return input('Do you want to play again? Enter Yes or No: ').lower().startswith('y')

while True:

clear\_output()

print('Welcome to Tic Tac Toe!')

toggle = random\_player()

player = players[toggle]

print('For this round, Player %s will go first!' %(player))

game\_on = True

input('Hit Enter to continue')

while game\_on:

display\_board(available,theBoard)

position = player\_choice(theBoard,player)

place\_marker(available,theBoard,player,position)

if win\_check(theBoard, player):

display\_board(available,theBoard)

print('Congratulations! Player '+player+' wins!')

game\_on = False

else:

if full\_board\_check(theBoard):

display\_board(available,theBoard)

print('The game is a draw!')

break

else:

toggle \*= -1

player = players[toggle]

clear\_output()

# reset the board and available moves list

theBoard = [' '] \* 10

available = [str(num) for num in range(0,10)]

if not replay():

break