Python- Tic Tac Toe Game

A function that can print out a board. Set up your board as a list, where each index 1-9 corresponds with a number on a number pad, so you get a 3 by 3 board representation.

Code:

```
from IPython.display import clear_output
def board(boardval):
    print(boardval[0]+' | '+boardval[1]+' | '+boardval[2])
    print('__|___|__')
    print(boardval[3]+' | '+boardval[4]+' | '+boardval[5])
    print('__|___|__')
    print(boardval[6]+' | '+boardval[7]+' | '+boardval[8])
finalboard=[' ']*10 #random list
board(finalboard)
```

A function that can take in a player input and assign their marker as 'X' or 'O'.

Code:

```
def players_input():
    player1=0
    player2=0
    while player1 is not 'X'or (player1 is not 'O'):
        player1=input("Player 1, select your representation : ")
        if player1=='X':
            return ('X','O')
            break
        elif player1=='O':
            return ('O','X')
            break
```

```
        else:
            continue
    print("\nPlayer 1 is "+player1+" and Player 2 is "+player2)
```

A function that takes in the board list object, a marker ('X' or 'O'), and a desired position (number 1-9) and assigns it to the board.

Code:

```
def place_marker(boardval,choice,position):
        boardval[position]=choice
```

A function that takes in a board and a mark (X or O) and then checks to see if that mark has won.

Code:

```
def winner_check(board, mark):
    while len(board)>5:
#runs the loop only when the length of board is greater than 5 because before that it's too early to declare a winner
        if (board[0]==board[1]==board[2]==mark) or (board[3]==board[4]==board[5]==mark) or (board[6]==board[7]==board[8]==mark):
#horizontal check
            print('The winner is '+mark)
            return True
        elif (board[0]==board[3]==board[6]==mark) or (board[1]==board[4]==board[7]==mark) or (board[2]==board[5]==board[8]==mark):
#vertical check
            print('The winner is '+mark)
            return True
        elif (board[0]==board[4]==board[8]==mark) or (board[2]==board[4]==board[6]==mark):
#diagonal check
            print('The winner is '+mark)
```

```
            return True
        else:
            print('No winner yet!!!')
            return False

        break
```

A function that uses the random module to randomly decide which player goes first.

Code:

```
import random
def firstchance():
    rand=random.randint(1,2)
    if rand==1:
        return 'Player 1'
    elif rand==2:
        return 'Player 2'
```

A function that returns a boolean indicating whether a space on the board is freely available.

Code:

```
def empty_space(board,position0):
    return board[position0]==' '
```

A function that checks if the board is full and returns a boolean value. True if full, False otherwise.

Code:

```
def board_check(board):
    count=0

    for items in range(0,9,):
```

```
        if board[items]=="X" or board[items]=="O":
            count+=1
    if count==9:
        return True
    else:
        return False
```

A function that asks for a player's next position (as a number 1-9) and then uses the function from step 6 to check if it's a free position. If it is, then return the position for later use.

Code:

```
def next_position(board):
    position1 = -1

    while position1 not in [0,1,2,3,4,5,6,7,8] or not empty_space(finalboard, position1):
        position1 = int(input('Choose your next position: (0-8) '))

    return position1
```

A function that asks the player if they want to play again and returns a boolean True if they do want to play again.

Code:

```
def play_again():
    playagain=input("Do you want to play again, Press Y or y : ")
    if playagain=='y' or playagain=='Y':
        return True
    else:
        return False
```

Arranging the functions to execute the game in accordance

```
import os
print("Welcome to the Game of Tic Tac Toe")
while True:
    finalboard=[' ']*10 #reseting the board to empty values at first

    #players_input() #players select their playing letter.
    player1,player2=players_input() #assigning thier respective symbols

    turnfirst=firstchance() #CPU desicison of which player goes first
    print(turnfirst+' will go first')

    while True:
        game_play=input('Are you ready to play, y or n : ').lower()
        if game_play=='y':
            game=True
            break
        elif game_play=='n':
            game=False
            False
        else:
            True

    while game:
        if turnfirst=='Player 1':
            os.system('clear')
            board(finalboard) #displays the board each time an input is given by the user
```

```
        positiongame=next_position(finalboard) #ask user for next position


        place_marker(finalboard,player1,positiongame) #fills the spot at given position by the
user
        if winner_check(finalboard,player1):
            print("Player 1 has won the game")
            board(finalboard)
            game=False
        else:
            if board_check(finalboard):
                board(finalboard)
                print("The game is a draw")
                break
            else:
                print("Chance: Player 2")
                turnfirst='Player 2'
    else:
        os.system('clear')
        board(finalboard) #displays the board each time an input is given by the user
        positiongame=next_position(finalboard) #ask user for next position


        place_marker(finalboard,player2,positiongame) #fills the spot at given position by the
user
        if winner_check(finalboard,player2):
            print("Player 2 has won the game")
            board(finalboard)
            game=False
        else:
            if board_check(finalboard):
```

```
            board(finalboard)
            print("The game is a draw")
            break
        else:
            print("Chance: Player 1")
            turnfirst='Player 1'

if not play_again(): #replay option
    False
```