import random

import tkinter

from tkinter import \*

from functools import partial

from tkinter import messagebox

from copy import deepcopy

# sign variable to decide the turn of which player

sign = 0

# Creates an empty board

global board

board = [[" " for x in range(3)] for y in range(3)]

# Check l(O/X) won the match or not

# according to the rules of the game

def winner(b, l):

    return ((b[0][0] == l and b[0][1] == l and b[0][2] == l) or

            (b[1][0] == l and b[1][1] == l and b[1][2] == l) or

            (b[2][0] == l and b[2][1] == l and b[2][2] == l) or

            (b[0][0] == l and b[1][0] == l and b[2][0] == l) or

            (b[0][1] == l and b[1][1] == l and b[2][1] == l) or

            (b[0][2] == l and b[1][2] == l and b[2][2] == l) or

            (b[0][0] == l and b[1][1] == l and b[2][2] == l) or

            (b[0][2] == l and b[1][1] == l and b[2][0] == l))

# Configure text on button while playing with another player

def get\_text(i, j, gb, l1, l2):

    global sign

    if board[i][j] == ' ':

        if sign % 2 == 0:

            l1.config(state=DISABLED)

            l2.config(state=ACTIVE)

            board[i][j] = "X"

        else:

            l2.config(state=DISABLED)

            l1.config(state=ACTIVE)

            board[i][j] = "O"

        sign += 1

        button[i][j].config(text=board[i][j])

    if winner(board, "X"):

        gb.destroy()

        box = messagebox.showinfo("Winner", "Player 1 won the match")

    elif winner(board, "O"):

        gb.destroy()

        box = messagebox.showinfo("Winner", "Player 2 won the match")

    elif(isfull()):

        gb.destroy()

        box = messagebox.showinfo("Tie Game", "Tie Game")

# Check if the player can push the button or not

def isfree(i, j):

    return board[i][j] == " "

# Check the board is full or not

def isfull():

    flag = True

    for i in board:

        if(i.count(' ') > 0):

            flag = False

    return flag

# Create the GUI of game board for play along with another player

def gameboard\_pl(game\_board, l1, l2):

    global button

    button = []

    for i in range(3):

        m = 3+i

        button.append(i)

        button[i] = []

        for j in range(3):

            n = j

            button[i].append(j)

            get\_t = partial(get\_text, i, j, game\_board, l1, l2)

            button[i][j] = Button(

                game\_board, bd=5, command=get\_t, height=4, width=8)

            button[i][j].grid(row=m, column=n)

    game\_board.mainloop()

# Decide the next move of system

def pc():

    possiblemove = []

    for i in range(len(board)):

        for j in range(len(board[i])):

            if board[i][j] == ' ':

                possiblemove.append([i, j])

    move = []

    if possiblemove == []:

        return

    else:

        for let in ['O', 'X']:

            for i in possiblemove:

                boardcopy = deepcopy(board)

                boardcopy[i[0]][i[1]] = let

                if winner(boardcopy, let):

                    return i

        corner = []

        for i in possiblemove:

            if i in [[0, 0], [0, 2], [2, 0], [2, 2]]:

                corner.append(i)

        if len(corner) > 0:

            move = random.randint(0, len(corner)-1)

            return corner[move]

        edge = []

        for i in possiblemove:

            if i in [[0, 1], [1, 0], [1, 2], [2, 1]]:

                edge.append(i)

        if len(edge) > 0:

            move = random.randint(0, len(edge)-1)

            return edge[move]

# Configure text on button while playing with system

def get\_text\_pc(i, j, gb, l1, l2):

    global sign

    if board[i][j] == ' ':

        if sign % 2 == 0:

            l1.config(state=DISABLED)

            l2.config(state=ACTIVE)

            board[i][j] = "X"

        else:

            button[i][j].config(state=ACTIVE)

            l2.config(state=DISABLED)

            l1.config(state=ACTIVE)

            board[i][j] = "O"

        sign += 1

        button[i][j].config(text=board[i][j])

    x = True

    if winner(board, "X"):

        gb.destroy()

        x = False

        box = messagebox.showinfo("Winner", "Player won the match")

    elif winner(board, "O"):

        gb.destroy()

        x = False

        box = messagebox.showinfo("Winner", "Computer won the match")

    elif(isfull()):

        gb.destroy()

        x = False

        box = messagebox.showinfo("Tie Game", "Tie Game")

    if(x):

        if sign % 2 != 0:

            move = pc()

            button[move[0]][move[1]].config(state=DISABLED)

            get\_text\_pc(move[0], move[1], gb, l1, l2)

# Create the GUI of game board for play along with system

def gameboard\_pc(game\_board, l1, l2):

    global button

    button = []

    for i in range(3):

        m = 3+i

        button.append(i)

        button[i] = []

        for j in range(3):

            n = j

            button[i].append(j)

            get\_t = partial(get\_text\_pc, i, j, game\_board, l1, l2)

            button[i][j] = Button(

                game\_board, bd=5, command=get\_t, height=4, width=8)

            button[i][j].grid(row=m, column=n)

    game\_board.mainloop()

# Initialize the game board to play with system

def withpc(game\_board):

    game\_board.destroy()

    game\_board = Tk()

    game\_board.title("Tic Tac Toe")

    l1 = Button(game\_board, text="Player : X", width=10)

    l1.grid(row=1, column=1)

    l2 = Button(game\_board, text = "Computer : O",

                width = 10, state = DISABLED)

    l2.grid(row = 2, column = 1)

    gameboard\_pc(game\_board, l1, l2)

# Initialize the game board to play with another player

def withplayer(game\_board):

    game\_board.destroy()

    game\_board = Tk()

    game\_board.title("Tic Tac Toe")

    l1 = Button(game\_board, text = "Player 1 : X", width = 10)

    l1.grid(row = 1, column = 1)

    l2 = Button(game\_board, text = "Player 2 : O",

                width = 10, state = DISABLED)

    l2.grid(row = 2, column = 1)

    gameboard\_pl(game\_board, l1, l2)

# main function

def play():

    menu = Tk()

    menu.geometry("250x250")

    menu.title("Tic Tac Toe")

    wpc = partial(withpc, menu)

    wpl = partial(withplayer, menu)

    head = Button(menu, text = "---Welcome to tic-tac-toe---",

                  activeforeground = 'red',

                  activebackground = "yellow", bg = "red",

                  fg = "yellow", width = 500, font = 'summer', bd = 5)

    B1 = Button(menu, text = "Single Player", command = wpc,

                activeforeground = 'red',

                activebackground = "yellow", bg = "red",

                fg = "yellow", width = 500, font = 'summer', bd = 5)

    B2 = Button(menu, text = "Multi Player", command = wpl, activeforeground = 'red',

                activebackground = "yellow", bg = "red", fg = "yellow",

                width = 500, font = 'summer', bd = 5)

    B3 = Button(menu, text = "Exit", command = menu.quit, activeforeground = 'red',

                activebackground = "yellow", bg = "red", fg = "yellow",

                width = 500, font = 'summer', bd = 5)

    head.pack(side = 'top')

    B1.pack(side = 'top')

    B2.pack(side = 'top')

    B3.pack(side = 'top')

    menu.mainloop()

# Call main function

if \_\_name\_\_ == '\_\_main\_\_':

    play()