SARASWATHI B

1BM19CS032

1. IMPLEMENT TIC-TAC-TOE GAME.

import random

import time

import sys

n=[j for j in range(1,10)]

board=[" "for i in range(9)]

print("Welcome to the game TIC-TAC-TOE")

pt=[]

db=[]

def print\_board():

    row1="| {} | {} | {} |".format(board[0],board[1],board[2])

    row2="| {} | {} | {} |".format(board[3],board[4],board[5])

    row3="| {} | {} | {} |".format(board[6],board[7],board[8])

    print(row1)

    print(row2)

    print(row3)

    print()

def player\_move(icon):

    if icon=="X":

        number=1

    elif icon=="O":

        number=2

    print("your turn player {}".format(number))

    try:

        choice1=int(input("Enter your move(1-9): ").strip())

        pt.append(choice1)

        if choice1>0 and choice1<=9:

                if board[choice1-1]==" ":

                    board[choice1-1]=icon

                else:

                    print()

                    print("This space was taken...:(")

                    print()

                    player\_move(icon)

        else:

                print("invalid choice... please enter again")

                player\_move(icon)

    except ValueError:

        player\_move(icon)

def cturn():

        print("Computer turn: ")

        time.sleep(0.5)

        print(n1)

        time.sleep(1)

def ai():

    global n1

    if board[n1-1]==" ":

        cturn()

    else:

        n1=random.choice(n)

        if n1 not in db:

            if n1 not in pt:

                db.append(n1)

                cturn()

            else:

                if len(pt)!=9:

                    ai()

                else:

                    pass

        else:

            if len(db)!=9:

                ai()

def player\_movec(icon):

    global n1

    def dup():

        global n1

        if(board[0]==icon and board[1]==icon) or (board[5]==icon and board[8]==icon) or (board[4]==icon and board[6]==icon):

            n1=3

            ai()

        elif(board[0]==icon and board[2]==icon) or (board[4]==icon and board[7]==icon):

            n1=2

            ai()

        elif(board[1]==icon and board[2]==icon) or (board[3]==icon and board[6]==icon) or (board[4]==icon and board[8]==icon):

            n1=1

            ai()

        elif(board[3]==icon and board[4]==icon) or (board[2]==icon and board[8]==icon):

            n1=6

            ai()

        elif(board[3]==icon and board[5]==icon) or (board[1]==icon and board[7]==icon) or (board[0]==icon and board[8]==icon) or\

             (board[2]==icon and board[6]==icon):

            n1=5

            ai()

        elif(board[4]==icon and board[5]==icon) or (board[0]==icon and board[6]==icon):

            n1=4

            ai()

        elif(board[6]==icon and board[7]==icon) or (board[2]==icon and board[5]==icon) or (board[0]==icon and board[4]==icon):

            n1=9

            ai()

        elif(board[6]==icon and board[8]==icon) or (board[1]==icon and board[4]==icon):

            n1=8

            ai()

        elif(board[7]==icon and board[8]==icon) or (board[0]==icon and board[3]==icon) or (board[2]==icon and board[4]==icon):

            n1=7

            ai()

        else:

            n1=random.choice(n)

            if n1 not in db:

                if n1 not in pt:

                    db.append(n1)

                    cturn()

                else:

                    if len(pt)!=9:

                        dup()

                    else:

                        pass

            else:

                if len(db)!=9:

                    dup()

                else:

                    pass

    dup()

    if board[n1-1]==" ":

        board[n1-1]=icon

    else:

        print()

        print("This space was taken...:(")

        print()

        player\_movec(icon)

def is\_victory(icon):

    if(board[0]==icon and board[1]==icon and board[2]==icon)or\

      (board[3]==icon and board[4]==icon and board[5]==icon)or\

      (board[6]==icon and board[7]==icon and board[8]==icon)or\

      (board[0]==icon and board[3]==icon and board[6]==icon)or\

      (board[1]==icon and board[4]==icon and board[7]==icon)or\

      (board[2]==icon and board[5]==icon and board[8]==icon)or\

      (board[0]==icon and board[4]==icon and board[8]==icon)or\

      (board[2]==icon and board[4]==icon and board[6]==icon):

        return True

    else:

        return False

def is\_draw():

    if " " not in board:

        return True

    else:

        return False

def game():

    ch=int(input("which mode you want to play\n1. computer vs player\n2. player vs player\nchoice: "))

    if ch==2:

        while True:

            print\_board()

            player\_move("X")

            print\_board()

            if is\_victory("X"):

                print("X wins.. :) Congratulations")

                sys.exit()

            elif is\_draw():

                print("It's a draw!")

                sys.exit()

            player\_move("O")

            if is\_victory("O"):

                print\_board()

                print("O wins.. :) Congratulations")

                sys.exit()

            elif is\_draw():

                print("It's a draw!")

                sys.exit()

    elif ch==1:

           while True:

                print\_board()

                player\_move("X")

                print\_board()

                if is\_victory("X"):

                    print("Player(X) wins.. :) Congratulations")

                    sys.exit()

                elif is\_draw():

                    print("It's a draw!")

                    sys.exit()

                player\_movec("O")

                if is\_victory("O"):

                    print\_board()

                    print("Computer(O) wins.. :) Congratulations")

                    sys.exit()

                elif is\_draw():

                    print("It's a draw!")

                    sys.exit()

game()

