**NAME:** Abdullah Aslam Qazi

**REGNO:** BCS 203263

Min Max Algo

CODE:

import tkinter as tk  
from tkinter import messagebox  
  
  
board = [[' ' for \_ in range(3)] for \_ in range(3)]  
player = 'X'  
game\_over = False  
  
window = tk.Tk()  
window.title("Tic-Tac-Toe")  
  
  
def handle\_click(row, col):  
 global player, game\_over  
  
 if board[row][col] == ' ' and not game\_over:  
  
 board[row][col] = player  
  
 buttons[row][col].config(text=player)  
  
  
 if check\_winner(player):  
 messagebox.showinfo("Game Over", f"Player {player} wins!")  
 game\_over = True  
  
 elif check\_tie():  
 messagebox.showinfo("Game Over", "It's a tie!")  
 game\_over = True  
 else:  
  
 player = 'O' if player == 'X' else 'X'  
  
 if player == 'O':  
 make\_computer\_move()  
  
  
def make\_computer\_move():  
 best\_score = float('-inf')  
 best\_move = None  
  
 for i in range(3):  
 for j in range(3):  
 if board[i][j] == ' ':  
 board[i][j] = 'O'  
 score = minimax(board, 0, False)  
 board[i][j] = ' '  
  
  
 if score > best\_score:  
 best\_score = score  
 best\_move = (i, j)  
  
 row, col = best\_move  
 board[row][col] = 'O'  
 buttons[row][col].config(text='O')  
  
 if check\_winner('O'):  
 messagebox.showinfo("Game Over", "Computer wins!")  
 global game\_over  
 game\_over = True  
  
 elif check\_tie():  
 messagebox.showinfo("Game Over", "It's a tie!")  
 game\_over = True  
  
*# Minimax algorithm*def minimax(board, depth, is\_maximizing):  
 scores = {'X': -1, 'O': 1, 'tie': 0}  
  
 if check\_winner('X'):  
 return scores['X']  
 elif check\_winner('O'):  
 return scores['O']  
 elif check\_tie():  
 return scores['tie']  
  
 if is\_maximizing:  
 best\_score = float('-inf')  
 for i in range(3):  
 for j in range(3):  
 if board[i][j] == ' ':  
 board[i][j] = 'O'  
 score = minimax(board, depth + 1, False)  
 board[i][j] = ' '  
 best\_score = max(score, best\_score)  
 return best\_score  
 else:  
 best\_score = float('inf')  
 for i in range(3):  
 for j in range(3):  
 if board[i][j] == ' ':  
 board[i][j] = 'X'  
 score = minimax(board, depth + 1, True)  
 board[i][j] = ' '  
 best\_score = min(score, best\_score)  
 return best\_score  
  
def check\_winner(player):  
  
 for i in range(3):  
 if board[i][0] == board[i][1] == board[i][2] == player:  
 return True  
 for j in range(3):  
 if board[0][j] == board[1][j] == board[2][j] == player:  
 return True  
  
 if board[0][0] == board[1][1] == board[2][2] == player or board[0][2] == board[1][1] == board[2][0] == player:  
 return True  
 return False  
  
  
def check\_tie():  
 for row in board:  
 if ' ' in row:  
 return False  
 return True  
  
  
buttons = []  
for i in range(3):  
 row = []  
 for j in range(3):  
 button = tk.Button(window, text=' ', font=('Arial', 20), width=5, height=2,  
 command=lambda i=i, j=j: handle\_click(i, j))  
 button.grid(row=i, column=j)  
 row.append(button)  
 buttons.append(row)  
  
  
window.mainloop()

OUTPUT:

