|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  | | --- | --- | --- | | ***Karachi Institute of Economics and Technology –North Nazimabad Site*** | | | | ***Course:***  ***Artificial Intelligence LAB*** | | | | ***Faculty: Akram Hussain*** | ***Due Date*** | **10 April, 2023** | | ***Class ID: 111703*** | ***Total Marks:*** | **10** | | ***Assignment # 01*** | ***Date: 26 March, 2023*** | | |

Student Name:  **Farrukh Ahmed Khan** Student ID: **11960**

**Q1. Develop** the Tic-Tac-Toe game (with two players and GUI based) using MinMax algorithm. [CLO: C3]

import tkinter as tk

def AnalyzBoard(board):

for i in range(0, 9, 3):

if board[i] == board[i+1] == board[i+2] and board[i] != 0:

return board[i]

for i in range(3):

if board[i] == board[i+3] == board[i+6] and board[i] != 0:

return board[i]

if board[0] == board[4] == board[8] and board[0] != 0:

return board[0]

if board[2] == board[4] == board[6] and board[2] != 0:

return board[2]

return 0

board = [0] \* 9

player1 = -1

player2 = 1

root = tk.Tk()

root.title("Tic Tac Toe")

buttons = []

for i in range(9):

button = tk.Button(root, width=10, height=5, font=('Helvetica', 20), command=lambda i=i: ButtonClick(i))

button.grid(row=i // 3, column=i % 3)

buttons.append(button)

turn\_label = tk.Label(root, text="Player 1's turn (X)", font=('Helvetica', 20))

turn\_label.grid(row=3, column=0, columnspan=3)

def ComputerPlayer(board):

best\_score = float('-inf')

best\_move = None

for i in range(9):

if board[i] == 0:

board[i] = player2

score = MinMax(board, False)

board[i] = 0

if score > best\_score:

best\_score = score

best\_move = i

return best\_move

def MinMax(board, is\_maximizing):

winner = AnalyzBoard(board)

if winner != 0:

if winner == player2:

return 1

else:

return -1

elif all(square != 0 for square in board):

return 0

if is\_maximizing:

best\_score = float('-inf')

for i in range(9):

if board[i] == 0:

board[i] = player2

score = MinMax(board, False)

board[i] = 0

best\_score = max(best\_score, score)

return best\_score

else:

best\_score = float('inf')

for i in range(9):

if board[i] == 0:

board[i] = player1

score = MinMax(board, True)

board[i] = 0

best\_score = min(best\_score, score)

return best\_score

def EndGame(winner):

global board

global buttons

global turn\_label

# disable all buttons

for button in buttons:

button.configure(state='disabled')

# display the winner or a tie message

if winner == player1:

message = "Player 1 (X) wins!"

elif winner == player2:

message = "Player 2 (O) wins!"

else:

message = "It's a tie!"

turn\_label.configure(text=message)

def ButtonClick(i):

global board

global player1

global player2

if board[i] != 0:

return

if turn\_label['text'] == "Player 1's turn (X)":

buttons[i]['text'] = "X"

board[i] = player1

turn\_label['text'] = "Player 2's turn (O)"

else:

buttons[i]['text'] = "O"

board[i] = player2

turn\_label['text'] = "Player 1's turn (X)"

winner = AnalyzBoard(board)

if winner != 0:

EndGame(winner)

return

if all(square != 0 for square in board):

EndGame(None)

return

if turn\_label['text'] == "Player 2's turn (O)":

move = ComputerPlayer(board)

ButtonClick(move)

root.mainloop()

