**Name :** Atharva Arjun Goral

**Roll No :** 3101033

**Batch :** B  **Div :** A

# **Assignment No. 2**

**CODE :**

import random

PLAYER = 'X'

COMPUTER = 'O'

EMPTY = ' '

def check\_winner(board, symbol):

# Check rows, columns, and diagonals for a win

for i in range(3):

if all(board[i][j] == symbol for j in range(3)) or all(board[j][i] == symbol for j in range(3)):

return True

return board[0][0] == board[1][1] == board[2][2] == symbol or board[0][2] == board[1][1] == board[2][0] == symbol

def get\_available\_moves(board):

return [(i, j) for i in range(3) for j in range(3) if board[i][j] == EMPTY]

def minimax(board, is\_maximizing):

if check\_winner(board, COMPUTER):

return 10

if check\_winner(board, PLAYER):

return -10

if not get\_available\_moves(board):

return 0

best\_score = -float('inf') if is\_maximizing else float('inf')

for i, j in get\_available\_moves(board):

board[i][j] = COMPUTER if is\_maximizing else PLAYER

score = minimax(board, not is\_maximizing)

board[i][j] = EMPTY

best\_score = max(best\_score, score) if is\_maximizing else min(best\_score, score)

return best\_score

def find\_best\_move(board):

best\_score = -float('inf')

best\_move = None

for i, j in get\_available\_moves(board):

board[i][j] = COMPUTER

score = minimax(board, False)

board[i][j] = EMPTY

if score > best\_score:

best\_score = score

best\_move = (i, j)

return best\_move

def print\_board(board):

for row in board:

print(" | ".join(row))

print("-" \* 5)

def play\_game():

board = [[EMPTY] \* 3 for \_ in range(3)]

print("Welcome to Tic Tac Toe!")

print\_board(board)

while True:

# Player move

try:

row, col = map(int, input("Enter your move (row col) [0-2]: ").split())

if board[row][col] != EMPTY:

print("Invalid move, try again.")

continue

except (ValueError, IndexError):

print("Invalid input, enter two numbers between 0 and 2.")

continue

board[row][col] = PLAYER

print\_board(board)

if check\_winner(board, PLAYER):

print("You win!")

break

if not get\_available\_moves(board):

print("It's a tie!")

break

# Computer move

print("Computer's move:")

move = find\_best\_move(board)

if move:

board[move[0]][move[1]] = COMPUTER

print\_board(board)

if check\_winner(board, COMPUTER):

print("Computer wins!")

break

if not get\_available\_moves(board):

print("It's a tie!")

break

if \_\_name\_\_ == "\_\_main\_\_":

play\_game()

**OUTPUT :**

Welcome to Tic Tac Toe!

| |

-----

| |

-----

| |

-----

Enter your move (row col) [0-2]: 0 0

X | |

-----

| |

-----

| |

-----

Computer's move:

X | |

-----

| O |

-----

| |

-----

Enter your move (row col) [0-2]: 0 2

X | | X

-----

| O |

-----

| |

-----

Computer's move:

X | O | X

-----

| O |

-----

| |

-----

Enter your move (row col) [0-2]: 2 1

X | O | X

-----

| O |

-----

| X |

-----

Computer's move:

X | O | X

-----

O | O |

-----

| X |

-----

Enter your move (row col) [0-2]: 1 2

X | O | X

-----

O | O | X

-----

| X |

-----

Computer's move:

X | O | X

-----

O | O | X

-----

| X | O

-----

Enter your move (row col) [0-2]: 2 0

X | O | X

-----

O | O | X

-----

X | X | O

-----

It's a tie!