**Experiment 6 (A.I)**

**Code:**

def minimax(brd, depth, is\_maximizing):

if check\_winner(brd, 'O'):

return 1

elif check\_winner(brd, 'X'):

return -1

elif is\_full(brd):

return 0

if is\_maximizing:

best\_score = -math.inf

for i in range(9):

if brd[i] == ' ':

brd[i] = 'O'

score = minimax(brd, depth + 1, False)

brd[i] = ' '

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

return best\_score

else:

best\_score = math.inf

for i in range(9):

if brd[i] == ' ':

brd[i] = 'X'

score = minimax(brd, depth + 1, True)

brd[i] = ' '

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

return best\_score

# Function for AI to make a move

def ai\_move():

best\_score = -math.inf

move = -1

for i in range(9):

if board[i] == ' ':

board[i] = 'O'

score = minimax(board, 0, False)

board[i] = ' '

if score > best\_score:

best\_score = score

move = i

board[move] = 'O'

# Main game loop

def main():

print("Welcome to Tic Tac Toe!")

print\_board()

while True:

player\_move()

print\_board()

if check\_winner(board, 'X'):

print("You win!")

break

elif is\_full(board):

print("It's a draw!")

break

print("AI's turn:")

ai\_move()

print\_board()

if check\_winner(board, 'O'):

print("AI wins!")

break

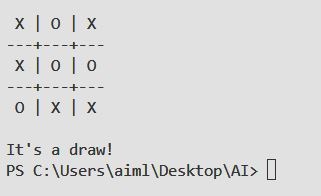
elif is\_full(board):

print("It's a draw!")

break

main()

**Output:**

****