Exp-13 minmax algorithm for gaming

Code

def is\_win(b, p):

wins = [(0,1,2),(3,4,5),(6,7,8),(0,3,6),(1,4,7),(2,5,8),(0,4,8),(2,4,6)]

return any(all(b[i] == p for i in combo) for combo in wins)

def minimax(b, player):

if is\_win(b, 'O'): return 1

if is\_win(b, 'X'): return -1

if ' ' not in b: return 0

if player == 'O':

best = -float('inf')

for i in range(9):

if b[i] == ' ':

b[i] = 'O'

best = max(best, minimax(b, 'X'))

b[i] = ' '

return best

else:

best = float('inf')

for i in range(9):

if b[i] == ' ':

b[i] = 'X'

best = min(best, minimax(b, 'O'))

b[i] = ' '

return best

def best\_move(b):

move = -1

best = -float('inf')

for i in range(9):

if b[i] == ' ':

b[i] = 'O'

score = minimax(b, 'X')

b[i] = ' '

if score > best:

best = score

move = i

return move

board = [' '] \* 9

while True:

print('Board:', board[0:3], board[3:6], board[6:9])

human = int(input("Your move (0-8): "))

if board[human] != ' ':

print("Spot taken!")

continue

board[human] = 'X'

if is\_win(board, 'X'):

print("You win!", board)

break

if ' ' not in board:

print("Draw!", board)

break

ai = best\_move(board)

if ai == -1:

print("Draw!", board)

break

board[ai] = 'O'

if is\_win(board, 'O'):

print("AI wins!", board)

break

output

