import random

# Tahta boyutu

BOARD\_SIZE = 3

# Oyun tahtası

board = [[' '] \* BOARD\_SIZE for \_ in range(BOARD\_SIZE)]

# Kedi ve köpek sembolleri

CAT\_SYMBOL = 'C'

DOG\_SYMBOL = 'D'

# Kedinin hamlesini gerçekleştirir

def cat\_move():

best\_score = float('-inf')

best\_move = None

# Tahta üzerindeki boş hücreleri dener

for i in range(BOARD\_SIZE):

for j in range(BOARD\_SIZE):

if board[i][j] == ' ':

board[i][j] = CAT\_SYMBOL

score = minimax(board, 0, False)

board[i][j] = ' '

# En iyi hamleyi seçer

if score > best\_score:

best\_score = score

best\_move = (i, j)

# Kedinin en iyi hamlesini gerçekleştirir

if best\_move:

board[best\_move[0]][best\_move[1]] = CAT\_SYMBOL

# Köpeğin hamlesini gerçekleştirir

def dog\_move():

available\_moves = []

# Boş hücreleri bulur

for i in range(BOARD\_SIZE):

for j in range(BOARD\_SIZE):

if board[i][j] == ' ':

available\_moves.append((i, j))

# Köpeğin rastgele hamlesini gerçekleştirir

if available\_moves:

move = random.choice(available\_moves)

board[move[0]][move[1]] = DOG\_SYMBOL

# Tahta üzerindeki kazananı kontrol eder

def check\_winner():

for i in range(BOARD\_SIZE):

# Satırları kontrol eder

if board[i][0] == board[i][1] == board[i][2] != ' ':

return board[i][0]

# Sütunları kontrol eder

if board[0][i] == board[1][i] == board[2][i] != ' ':

return board[0][i]

# Çaprazları kontrol eder

if board[0][0] == board[1][1] == board[2][2] != ' ':

return board[0][0]

if board[0][2] == board[1][1] == board[2][0] != ' ':

return board[0][2]

# Berabere durumunu kontrol eder

if all(board[i][j] != ' ' for i in range(BOARD\_SIZE) for j in range(BOARD\_SIZE)):

return 'Berabere'

return None

# Minimax algoritması

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

winner = check\_winner()

if winner:

if winner == CAT\_SYMBOL:

return 1

elif winner == DOG\_SYMBOL:

return -1

else:

return 0

if is\_maximizing:

best\_score = float('-inf')

for i in range(BOARD\_SIZE):

for j in range(BOARD\_SIZE):

if board[i][j] == ' ':

board[i][j] = CAT\_SYMBOL

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(BOARD\_SIZE):

for j in range(BOARD\_SIZE):

if board[i][j] == ' ':

board[i][j] = DOG\_SYMBOL

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

board[i][j] = ' '

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

return best\_score

# Oyun döngüsü

def play\_game():

current\_player = CAT\_SYMBOL

while True:

if current\_player == CAT\_SYMBOL:

cat\_move()

current\_player = DOG\_SYMBOL

else:

dog\_move()

current\_player = CAT\_SYMBOL

print\_board()

winner = check\_winner()

if winner:

if winner == 'Berabere':

print("Oyun berabere bitti.")

else:

print("Kazanan:", winner)

break

# Tahtayı yazdırır

def print\_board():

for row in board:

print('|'.join(row))

print('-' \* (BOARD\_SIZE \* 2 - 1))

# Oyunu başlatır

def start\_game():

print("Kedi ve Köpek Oyununa Hoş Geldiniz!")

print("Tahta Boyutu:", BOARD\_SIZE, "x", BOARD\_SIZE)

print("Kedi sembolü:", CAT\_SYMBOL)

print("Köpek sembolü:", DOG\_SYMBOL)

print("Kedi oyuna başlıyor.")

print\_board()

play\_game()

# Oyunu başlatır

start\_game()