Bahria University, Islamabad

Department of Software Engineering

Artificial Intelligence Lab

(Fall-2021)

Teacher: Engr. M Waleed Khan

Student : M Iqrar Ijaz Malik

Enrollment : 01-131182-021

Lab Journal:9

Date: 10-11-2021

Task No:	Task Wise Marks		Documentation Marks		Total Marks (20)
	Assigned	Obtained	Assigned	Obtained	
1	15		5		

Comments:

Signature

LAB

Lab 9: Implementing AI Agents

Introduction

In artificial intelligence, an intelligent agent (IA) refers to an autonomous entity which acts, directing its activity towards achieving goals (i.e., it is an agent), upon an environment using observation through sensors and consequent actuators (i.e. it is intelligent). Intelligent agents may also learn or use knowledge to achieve their goals. They may be very simple or very complex. A reflex machine, such as a thermostat, is considered an example of an intelligent agent.

Classes:

1. simple reflex agents
2. model-based reflex agents
3. goal-based agents
4. utility-based agents
5. learning agents

Learning agents

Learning has the advantage that it allows the agents to initially operate in unknown environments and to become more competent than its initial knowledge alone might allow. The most important distinction is between the "learning element", which is responsible for making improvements, and the "performance element", which is responsible for selecting external actions.

Exercise:

Task 1

Implement and AI agent to solve 8 Puzzle Problem using A* Algorithm

1 vs 1 mode

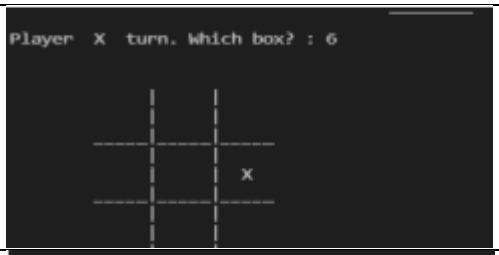
```
def print_tic_tac_toe(values):
    print("\n")
    print("\t  |  |")
    print("\t {} | {} | {}".format(values[0], values[1],
values[2]))
    print('\t_____|_____|_____')
    print("\t  |  |")
    print("\t {} | {} | {}".format(values[3], values[4],
values[5]))
    print('\t_____|_____|_____')
    print("\t  |  |")
    print("\t {} | {} | {}".format(values[6], values[7],
values[8]))
    print("\t  |  |")
    print("\n")

# Function to check if any player has won
def check_win(player_pos, cur_player):
    # All possible winning combinations
    soln = [[1, 2, 3], [4, 5, 6], [7, 8, 9], [1, 4, 7], [2, 5,
8], [3, 6, 9], [1,
    5, 9], [3, 5, 7]]
    # Loop to check if any winning combination is satisfied
    for x in soln:
        if all(y in player_pos[cur_player] for y in x):
        # Return True if any winning combination satisfies
            return True
    # Return False if no combination is satisfied
    return False
    # Function to check if the game is drawn
def check_draw(player_pos):
    if len(player_pos['X']) + len(player_pos['O']) == 9:
        return True
    return False

# Function for a single game of Tic Tac Toe
def single_game(cur_player):
    # Represents the Tic Tac Toe
    values = [' ' for x in range(9)]
    # Stores the positions occupied by X and O
    player_pos = {'X':[], 'O':[]}
    # Game Loop for a single game of Tic Tac Toe
    while True:
```

```
        print_tic_tac_toe(values)
        # Try exception block for MOVE input
        try:
            print("Player ", cur_player, " turn. Which box? :
", end="")
            move = int(input())
        except ValueError:
            print("Wrong Input!!! Try Again")
            continue

            # Sanity check for MOVE inout
        if move < 1 or move > 9:
            print("Wrong Input!!! Try Again")
        continue

        # Check if the box is not occupied already
        if values[move - 1] != ' ':
            print("Place already filled. Try again!!")
        continue

        # Update game information
        # Updating grid status
        values[move - 1] = cur_player
        # Updating player positions
        player_pos[cur_player].append(move)
        # Function call for checking win
        if check_win(player_pos, cur_player):
            print_tic_tac_toe(values)
            print("Player ", cur_player, " has won the
game!!")
            print("\n")
            return cur_player

        # Function call for checking draw game
        if check_draw(player_pos):
            print_tic_tac_toe(values)
        print("Game Drawn")
        print("\n")
        return 'D'

        # Switch player moves
        if cur_player == 'X':
            cur_player = 'O'
        else:
            cur_player = 'X'


if __name__ == "__main__":
    print("Player 1")
    player1 = input("Enter the name : ")
    print("\n")
```

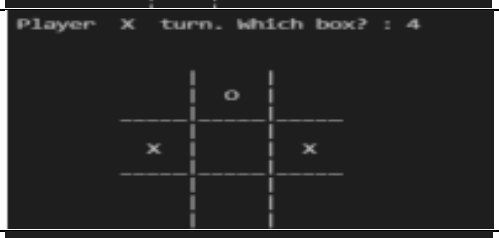
```
    print("Player 2")
    player2 = input("Enter the name : ")
    print("\n")
    # Stores the player who chooses X and O
    cur_player = player1
    # Stores the choice of players
    player_choice = {'X' : "", 'O' : ""}
    # Stores the options
    options = ['X', 'O']
    # Game Loop for a series of Tic Tac Toe
    # The loop runs until the players quit
    while True:
        # Player choice Menu
        print("Turn to choose for", cur_player)
        print("Enter 1 for X")
        print("Enter 2 for O")
        print("Enter 3 to Quit")
        # Try exception for CHOICE input

        choice = int(input())
        if choice == 1:
            player_choice['X'] = cur_player
            if cur_player == player1:
                player_choice['O'] = player2
            else:
                player_choice['O'] = player1
        elif choice == 2:
            player_choice['O'] = cur_player
            if cur_player == player1:
             player_choice['X'] = player2
            else:
                player_choice['X'] = player1

        elif choice == 3:
            print("Thank You")
            break

        else:
            print("Wrong Choice!!!! Try Again\n")
        # Stores the winner in a single game of Tic Tac Toe
        winner = single_game(options[choice - 1])
        # Switch player who chooses X or O
        if cur_player == player1:
            cur_player = player2
        else:
            cur_player = player1
```

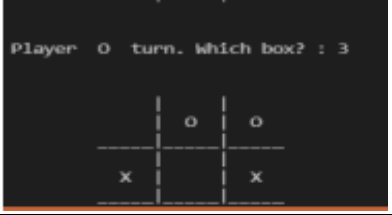
Output:

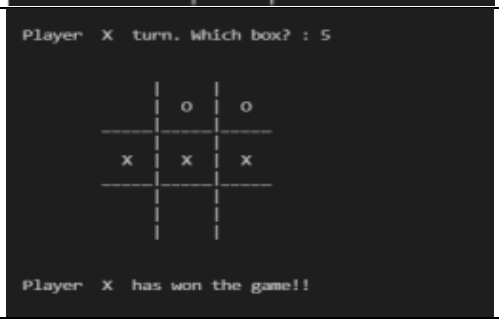

Enter 1 for X
Enter 2 for O
Enter 3 to Quit
1


Player X turn. Which box? : 6


Player O turn. Which box? : 2


Player X turn. Which box? : 4


Player O turn. Which box? : 3


Player X turn. Which box? : 5
Player X has won the game!!

1 vs Computer

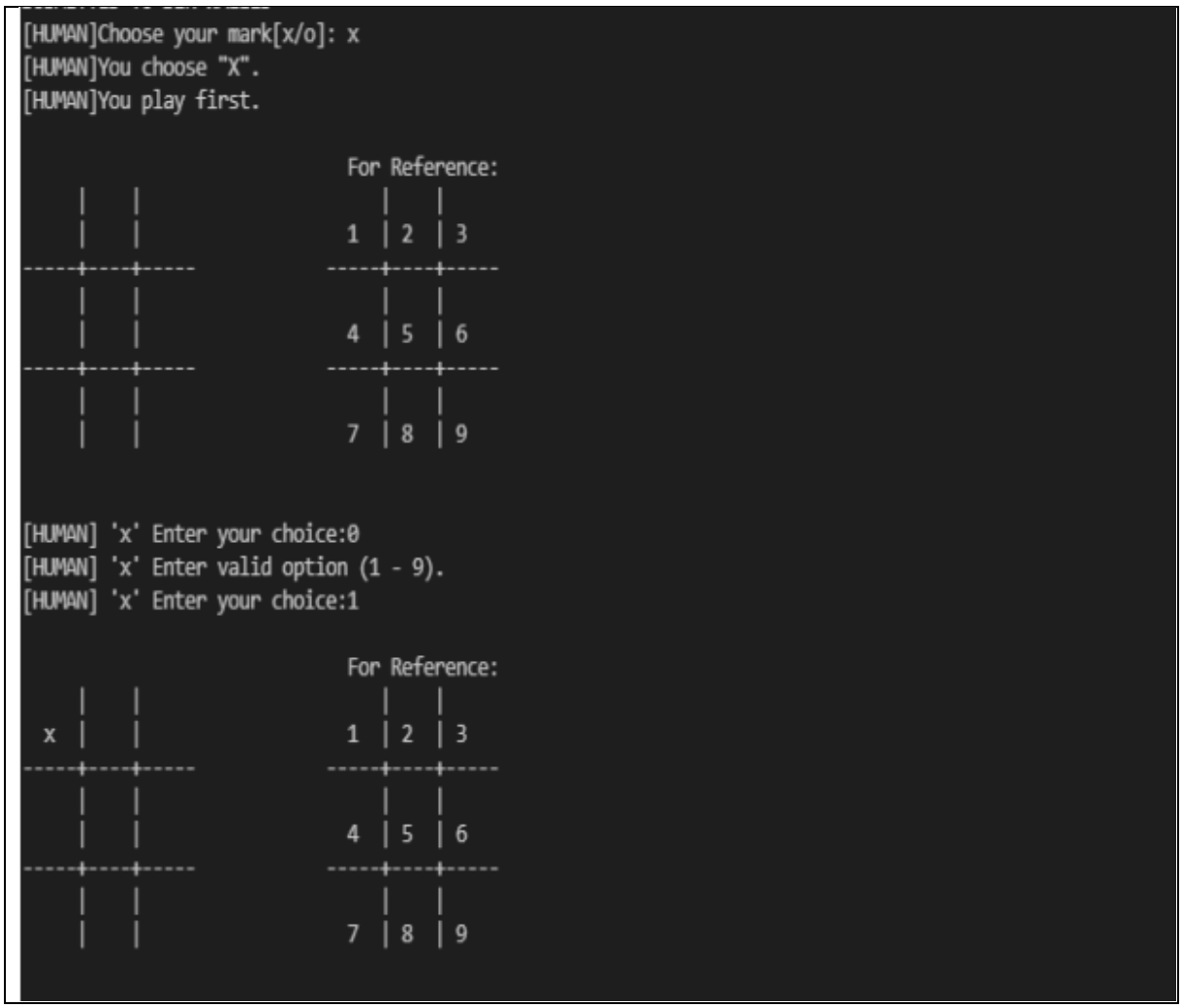
```
def display_board():
    print('    |   |   ',10*' ',' |   |   ',)
    print(' '+board[1]+' | '+board[2]+'  | '+board[3]+'
',10*' ',' 1 | 2 | 3 ')
    print('    +   +   ',10*' ',"  +   +   ")
    print('    |   |   ',10*' ',"  |   |   ")
    print(' '+board[4]+' | '+board[5]+'  | '+board[6]+'
',10*' ',"  4 | 5 | 6    ")
    print('    +   +   ',10*' ',"  +   +   ")
    print('    |   |   ',10*' ',"  |   |   ")
    print(' '+board[7]+' | '+board[8]+'  | '+board[9]+'
',10*' ',"  7 | 8 | 9    \n\n")


def human_input(mark):
        while True:
            inp = input(f"[HUMAN] '{mark}' Enter your
choice:")

    if inp.isdigit() and int(inp) < 10 and int(inp) > 0:
        inp = int(inp)
    if board[inp] == " ":
        return inp
    else:
        print(f"[HUMAN] '{mark}' place already taken.")
    else:
    print(f"[HUMAN] '{mark}' Enter valid option (1 - 9).")


def winning(mark,board):
    winning_place =
[[1,2,3],[4,5,6],[7,8,9],[1,4,7],[2,5,8],[3,6,9],[1,5,9],[3,5
,7]]
    for win_place in winning_place:
     if board[win_place[0]] == board[win_place[1]] ==
board[win_place[2]] ==mark:
                return True

     def cpu_input(cpu, human, board):
         for i in range(1, 10):
             if board[i] == ' ' and win_move(i, board, cpu):
             return i

     for i in range(1, 10):
         if board[i] == ' ' and win_move(i, board, human):
             return i
     for i in [5, 1, 7, 3, 2, 9, 8, 6, 4]:
         if board[i] == ' ':
             return i
```

```
def new_game():
        while True:
            nxt = input('[HUMAN] Do you want to play
again?(y/n):')

    if nxt in ['y', 'Y']:
        again = True
    break
    elif nxt in ['n', 'N']:
    print('Have a great day')
    again = False
    break
    else:
    print('Enter correct input')
    if again:
        print('__________NEW GAME__________')
    main_game()
    else:
        return False


def win_check(human , cpu):
    winning_place =

[[1,2,3],[4,5,6],[7,8,9],[1,4,7],[2,5,8],[3,6,9],[1,5,9],[3,5
,7]]
    for win_place in winning_place:
        if board[win_place[0]] == board[win_place[1]] ==
board[win_place[2]] == human:
            print('[HUMAN] wins the match!')
        if not new_game():
            return False
    elif board[win_place[0]] == board[win_place[1]] ==
board[win_place[2]] ==cpu:
            print('[CPU] wins the match!')
             if not new_game():
                 return False
        if ' ' not in board:
    print('MATCH DRAW!!')
                if not new_game():
            return False
    return True


def user_choice():
    while True:
        inp = input('[HUMAN]Choose your mark[x/o]: ')
            if inp in ['x' , 'X']:
             print('[HUMAN]You choose "X".\n[HUMAN]You play
first.')
```

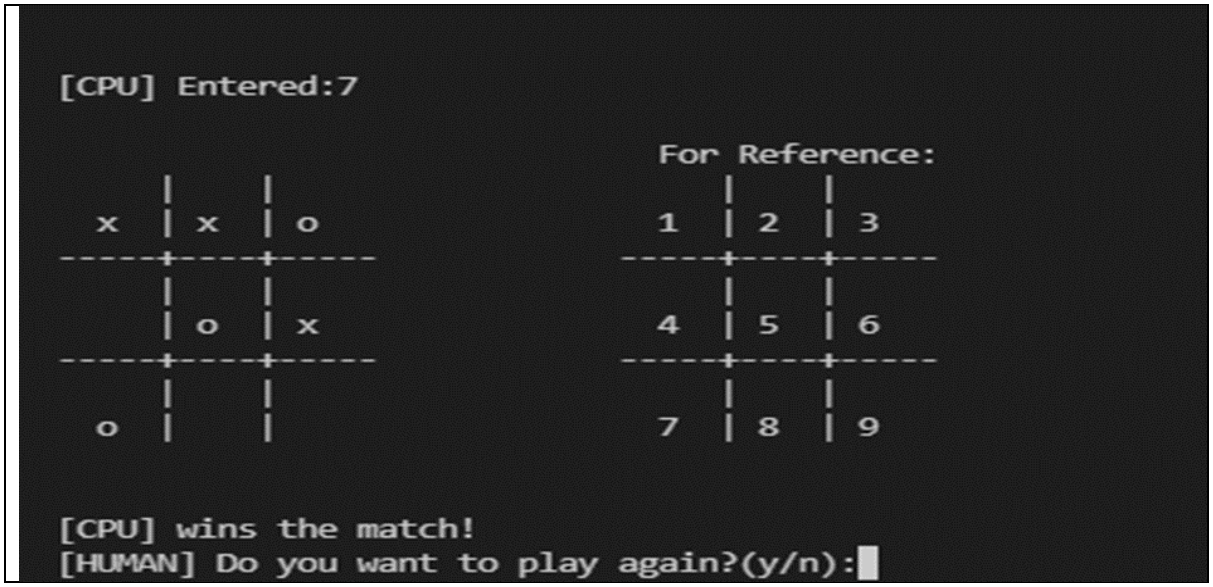
```
                    return 'x','o'
                elif inp in ['O','o']:
            print('[HUMAN] You choose "O".\n[HUMAN] CPU plays
first.')
             return 'o','x'
                else:
          print('[HUMAN] Enter correct input!')

def human_input(mark): while True:
    inp = input(f"[HUMAN] '{mark}' Enter your choice:") if
inp.isdigit() and int(inp) <10 and int(inp) >0:
         inp = int(inp)
     if board[inp] == " ": return inp
    else:
         print(f"[HUMAN] '{mark}' place already taken.")
    else:
         print(f"[HUMAN] '{mark}' Enter valid option (1 -
9).")


def winning(mark,board):
    winning_place =
[[1,2,3],[4,5,6],[7,8,9],[1,4,7],[2,5,8],[3,6,9],[1,5,9],[3,5
,7]]
    for win_place in winning_place:
             if board[win_place[0]] == board[win_place[1]] ==
board[win_place[2]] == mark:
                return True


def win_move(i,board,mark):
    temp_board = list(board) temp_board[i] = mark
         if winning(mark,temp_board): return True
             else:
                 return False



def main_game():
    global board
    play = True
    board =['',' ',' ',' ',' ',' ',' ',' ',' ',' ']
    human , cpu = user_choice()
    display_board()
    while play:
         if human == 'x':
             x = human_input(human)
             board[x] = human
             display_board()
             play = win_check(human, cpu)
             if play:
```

```
                o = cpu_input(cpu, human, board)
                print(f'[CPU] Entered:{o}')
                board[o] = cpu
            display_board()
            play = win_check(human, cpu)
            else:
            x = cpu_input(cpu, human, board)
            print(f'[CPU] Entered:{x}')
            board[x] = cpu
            display_board()
            play = win_check(human, cpu)
            if play:
                o = human_input(human)
                board[o] = human
                display_board()
            play = win_check(human, cpu)

            if name == ' main ': main_game()
```

Output:



```
[CPU] Entered:5

                                For Reference:
     |     |                         |     |
  x  |     |                      1  |  2  |  3
-----+-----+-----                -----+-----+-----
     |     |                         |     |
     |  o  |                      4  |  5  |  6
-----+-----+-----                -----+-----+-----
     |     |                         |     |
     |     |                      7  |  8  |  9


[HUMAN] 'x' Enter your choice:1
[HUMAN] 'x' place already taken.
[HUMAN] 'x' Enter your choice:2

                                For Reference:
     |     |                         |     |
  x  |  x  |                      1  |  2  |  3
-----+-----+-----                -----+-----+-----
     |     |                         |     |
     |  o  |                      4  |  5  |  6
-----+-----+-----                -----+-----+-----
     |     |                         |     |
     |     |                      7  |  8  |  9


[CPU] Entered:3

                                For Reference:
     |     |                         |     |
  x  |  x  |  o                   1  |  2  |  3
-----+-----+-----                -----+-----+-----
     |     |                         |     |
     |  o  |                      4  |  5  |  6
-----+-----+-----                -----+-----+-----
     |     |                         |     |
     |     |                      7  |  8  |  9


[HUMAN] 'x' Enter your choice:6

                                For Reference:
     |     |                         |     |
  x  |  x  |  o                   1  |  2  |  3
-----+-----+-----                -----+-----+-----
     |     |                         |     |
     |  o  |  x                   4  |  5  |  6
-----+-----+-----                -----+-----+-----
     |     |                         |     |
     |     |                      7  |  8  |  9
```



Conclusion

I completed the tasks given to us and pasted output above.