

Group Members:

1. Qui, Christian Michael
2. Regencia, Josiah Eleazar

Requirements:

1. Python2.7.* or Python3

The project is an implementation of uninformed search algorithms using a self-defined PEAS. The objective of this simulation is that in a 3 x 3 chess board, where a knight chess piece and an immobile enemy piece are randomly generated anywhere in the board other than [1,1] (because this is the center of the board, and it's impossible for the knight to visit or move from it.), the knight must search for the enemy piece. The knight piece must follow the traversal rules in the game of chess, where it must traverse 2 cells horizontally/vertically, and move an additional cell horizontally or vertically, depending on the former movement. The PEAS are as follows:

Performance measure – The knight reaches the cell where the enemy is located

Environment – 3 x 3 chessboard.

Actuator – “L” traversal of the knight.

Sensor – Knight knows its location, and its next moves.

Notes on how to run the Project:

1. Main file is knight_tour.py
2. Tree_traversal.py accepts an argument to select whether the program will run a dfs or a bfs.
 - a. To run bfs: python knight_tour.py bfs
 - b. To run dfs: python knight_tour.py dfs
3. The arguments 'dfs' or 'bfs' are not case-sensitive. So it can be DFS or Dfs.

Notes on the output:

1. Below is a sample output of BFS:

```
Josiahs-MacBook-Pro:project_1 jetregencia$ python knight_tour.py bfs
Knight Location: (0, 1)
Enemy Location: (0, 2)

current state: 0 1    goal state: 0 2
Number of possible moves: 2    Possible moves: (2,2) (2,0)

current state: 2 2    goal state: 0 2
Number of possible moves: 1    Possible moves: (1,0)

current state: 2 0    goal state: 0 2
Number of possible moves: 1    Possible moves: (1,2)

current state: 1 0    goal state: 0 2
Number of possible moves: 1    Possible moves: (0,2)

current state: 1 2    goal state: 0 2
Number of possible moves: 1    Possible moves: (0,0)

current state: 0 2    goal state: 0 2

Total Traverse Time: 0.000155925750732
Josiahs-MacBook-Pro:project_1 jetregencia$
```

2. Below is the sample output of DFS:

```
Josiahs-MacBook-Pro:project_1 jetregencia$ python knight_tour.py dfs
Knight Location: (0, 1)
Enemy Location: (1, 2)

current state: 0 1    goal state: 1 2
Number of possible moves: 2    Possible moves: (2,2) (2,0)

current state: 2 2    goal state: 1 2
Number of possible moves: 1    Possible moves: (1,0)

current state: 1 0    goal state: 1 2
Number of possible moves: 1    Possible moves: (0,2)

current state: 0 2    goal state: 1 2
Number of possible moves: 1    Possible moves: (2,1)

current state: 2 1    goal state: 1 2
Number of possible moves: 1    Possible moves: (0,0)

current state: 0 0    goal state: 1 2
Number of possible moves: 1    Possible moves: (1,2)

current state: 1 2    goal state: 1 2

Total Traverse Time: 0.000185012817383
Josiahs-MacBook-Pro:project_1 jetregencia$
```

3. In the first two lines, the randomly generated locations of the knight and the enemy are printed.
4. The lines that follow this are the traversals of the knight.
5. When the goal has been reached, total traverse time is printed, and the program stops.