Introduction to Artificial Intelligence

Program – Search in Pac-Man

Mar 10, 2025

Objectives

Practice and get familiar with the way to solve problem by searching. In this assignment you need to make use of the taught subject matters about Solving Problem by Searching (ch. 3) and Informed Search and Exploration (ch. 4).

Program

Write Python programs to solve the following questions of the [Search in Pac-Man] project indicated in the search.html.

Part A: Finding a Fixed Food Dot using Search Algorithms

- **Q1**: Use the depth-first search (DFS) algorithm.
- **Q2**: Use the breadth-first search (BFS) algorithm.
- **Q4**: Implement and test the A* search algorithm.

Part B: Finding All the Corners

- **Q5**: Use an uninformed search (e.g., BFS).
- **Q6**: Use the A* search algorithm and design a heuristic for it.

Report

- Explain your methods (e.g., design of heuristics) for Q4 and Q6.

Submission

- **2025/03/23 24:00 (degrade by 10 points for each day delay)**
- Upload the report and the source code (`search.py` and searchAgents.py) to eeclass.
  - Other files will be ignored during testing and scoring.
  - Remember to remove or comment out any debugging code snippets before submission.

Useful Information

- Read the search.html thoroughly
- Define efficient search state representation: An efficient representation can greatly reduce search space
- Make sure to use the Stack, Queue and PriorityQueue data structures provided in util.py
- Part A:
  - Implement the functions `depthFirstSearch()` and `depthFirstSearch()` in search.py
  - Read the comments before you start
  - Remove `util.raiseNotDefined()` before testing your code
  - Do NOT add/edit/delete anything outside of the above-mentioned functions
- Part B
  - Implement function `aStarSearch()` in search.py
  - Find class `CornersProblem` in searchAgents.py and implement
    - function `__init__( .. )`
    - function `getStartState(self)`
    - function `isGoalState(self, state)`
    - function `getSuccessors(self, state)`
  - Implement function `cornersHeuristic(state, problem)` in searchAgents.py
  - Read the comments before you start
  - Remove `util.raiseNotDefined()` before testing your code
  - Do NOT add/edit/delete anything outside of the above-mentioned functions
- Data structures like Queue, Stack, and Priority Queue along with their own `push()` and `pop()` methods are already implemented in utils.py
  - You can store all types of variable in them, such as `integer`, `string`, `list`, `tuple`, `dictionary`, ...
  - Example:
    ```
    candidates = util.Queue()
    candidates.push("apple")
    print(candidate.pop())
    ```
- The file commands.txt contains some testing commands you can play with
- To speed up your testing, append `--frameTime 0` to the command you are testing
- To downsize a game window, append `-z .5` to the command you are testing