2021/05/27

COSE361(03)-Final project 1/2

Pacman algorithm

화공생명공학과

2015170063

오원석

Submission

1. Submit myAgents.py and on Blackboard.

- 별도로 첨부하여 제출하였습니다.

2. Submit a 2 page report(pdf) file containing

1. capture the result of pacman.py with layout test71.lay (4 points)

2. Description of your agents. (2 points)

This agent is part of getAction, initialize, findPathToClosestDot, and isGoalState. The first getAction function is the skeleton of the most important part of my agent, and the initialize part is literally the initialization part. FindPathToClosestDot is the process of finding the nearest dot, and isGoalState shows what needs to be done when the last goalstate is reached.

3. Three discussions when playing Pacman 2 point for each discussion.(Total 6 points.)
-Discuss cases where the agent implemented by yourself is ***better*** than the baseline.

Baseline is a CLOSEST dot agent and you want to see if MyAgent results better than in this situation. Closest dot agent is when you eat a dot that is near greedy. It has difficulty eating dot effectively. For example, BFS Search is an advantage. In this case, the result is better than the base line, Closest dot agent.

-Discuss cases where the agent implemented by yourself is ***worse*** than the baseline.

This is the opposite of the above results. You can think of BfS's search as an unfavorable case. In this case, a DFS that has one more length advantage over a BFS that the agent is exploring the maze as a whole. This can be thought of as a case of DFS taking an advantageous path. If you set a Greedy case and eat dot, you may end up eating dot closer to you. If this case is continuous, it will result in an advantageous case in dfs exploration. Therefore, the baseline may be advantageous in this case.

-Ask & Answer your own question about the above discussion.

Comparison between Myagent and baseline in typical non-specific cases
We have previously compared BFS and DFS algorithm in Maze. My question is to compare the two agent in a typical case other than a particular Maze. Assuming that the Maze does not have a cycle, if any, the DFS can end as soon as it finds an exit and the length to that exit can be long. The BFS must visit all the compartments of the Maze and can find the shortest route to the exit. Each time complexity exactly matches the size of the maze. So Myagent is better.