Berkley Pacman Game using Python 2.7

Matthew Berkman

Overview/Abstract

The goal of this assignment is to have us practice and understand basic search algorithms that are used to have computers solve problems, such as a maze, with no knowledge of problem except what end goal is. Objective is to have us create a program where Pacman can complete different objectives by taking a optimal path and finding that path quickly.

Problem Statement

Question 1-4 involved us coding the algorithms depth first search, breadth first search, uniform cost search, and A\* search so that they can be used by provided function PositionSearchProblem in searchAgents.py to have Pacman traverse a maze completing provided objective which in this case was getting to 1 food pellet.

Question 5 involved us implementing search problem CornersProblem which involved Pacman reaching 4 ccorners instead of just one location like in PositionSearchProblem.

Question 6 involved creating a Heuristic for Corners problem so that Pacman can find a solution through mazes faster while still finding most optimal path.

Question 7 involved creating a heuristic for provided food search problem which involves Pacman finding most optimal path to eat all the foods.

Solution Design

For question 1-4 created a function called graphSearch which is a basic search function which is configured by the different queuing strategies passed into it by the different search algorithms.

Depth First Search uses a stack

Breadth First Search uses a queue

Uniform Cost Search uses a priority queue

A\* search uses a priority queue along with combination of uniform-cost and greedy algorithms to calculate cost.

For question 5, mimicked Position Search Problem but adjusted it to where it takes into consideration multiple goals and finds a path to each of them

For question 6, wrote a heuristic that finds an optimal path without using to much resources for corner search problem.

For question 7, wrote a heuristic that finds an optimal path for having Pacman eat all the food in the maze, if possible.

Solution Implementation

[wasn’t exactly sure what you wanted here]

For question 1-4

Had each search algorithm call the basic search algorithm graphSearch and pass in their own queuing strategy so that in can perform the algorithm type search. Once graphSearch reached end goal, it returned a list of directions that Pacman needs to take tog et from starting position to single goal.

Question 5-7

Followed instructions of how to test code and made sure right functions were called and it was doing calculations properly.

Solution Result/Evaluation

Got full points on auto grader and most of the code ran quickly except for heuristic for eating all the pellets. Though since I used provided functions to help in my implementation, couldn’t do much since didn’t want to edit code that wasn’t part of the assignment.

Conclusion/Reflection

The main challenge I feel of this assignment was testing since we had to use Python 2.7 and used IDE eclipse which had a python feature which had a buggy debug tool. Had to print things to console to trace code. Though this was a very educational assignment that had us explore basics of A.I and how they function. Made me realize I need to review more algorithms so in future assignments I can write solutions for different types of problem at a faster rate.