Project – Phase C

Introduction to AI – CS487, Fall 2023

Deadline: 24/01/2024

Overview

This project is originally designed at Berkeley in order for the students to apply an array of AI techniques to playing Pacman. You will explore different techniques such as informed/uninformed search, multi-agent problems, reflex agents, probabilistic inference etc. This project allows students to visualize the results of the techniques they implement. They also contain code examples and clear directions, but do not force students to wade through undue amounts of scaffolding. Finally, Pacman provides a challenging problem environment that demands creative solutions.

Introduction

In this phase, you need to create an agent that is able to solve the original map of Pacman in relatively short time. This phase is a competition between you and your fellow students.

Files to Edit and Submit: Any files you alter or add in the folder provided in Phase B

Academic Dishonesty: We will be checking your code against other sub-missions in the class for logical redundancy. If you copy someone else's code and submit it with minor changes, we will know. These cheat detectors are quite hard to fool, so please don't try. We trust you all to submit your own work only; please don't let us down.

Description

Create an agent (easiest way to do it is to alter your Expectimax agent from the previous phase) that is able to solve the original map of pacman, achieving the highest score possible in the minimum amount of time. In order to evaluate your implementation, run the originalClassic layout 10 times and take the average run time and score achieved. Report it in a PDF file where you will also explain your solution. If your implementation is fast enough, the graphics should not be laggy. In the end of the project there will be a leaderboard with the highest scores achieved amongst all of the students. The students with the 3 best performing agents will be granted a bonus in the final grade.
You need to define the function getAction(gamestate) in your agent. In case you alter the expectimax agent, you can use the following command to test it:

```
pacman.py -p ExpectimaxAgent -a depth=3 -l originalClassic
```