Project 2: Multi-Agent Search in Pacman

CS 4300: Artificial Intelligence
University of Utah

In this project, you will design agents for the classic version of Pacman, including ghosts. Along the way, you will implement both minimax and expectimax search and try your hand at evaluation function design.

The code base has not changed much from the previous project, but please start with a fresh installation, rather than intermingling files from project 1.

As in project 1, this project includes an autograder for you to grade your answers on your machine. This can be run on all questions with the command:

python autograder.py

The code for this project consists of several Python files, some of which you will need to read and understand in order to complete the assignment, and some of which you can ignore. Download p02.zip which will contain all the code and supporting files.

1 Files to edit

For all the problems in this project, you need to edit just one python file, namely:

- **multiAgents.py:** where all of your multi-agent search agents will reside.

2 Supporting Files

The following python files will help you in understanding the problem and will get you familiar with the different data structures and games states in Pacman.

- **pacman.py:** The main file that runs Pacman games. This file describes a Pacman GameState type, which you use in this project.
- **game.py:** The logic behind how the Pacman world works. This file describes several supporting types like AgentState, Agent, Direction, and Grid.
- **util.py:** Useful data structures for implementing search algorithms.

3 Multi-Agent Search in Pacman (30 pts)

For all the problem titles described below, please refer to this multi-agent project page for the problem description and what is expected of each problem. As always autograder has different test cases against which you can run your program to check the correctness. For

the questions asked below, please ensure your response is brief and to the point. Please don't write paragraphs of text as responses to these questions.

3.1 Reflex Agent (4.5 pts)

1. (4 pts) Code implementation
2. (0.5 pt) What feature (or features) did you use for your evaluation function?

3.2 Minimax (5.5 pts)

1. (5 pts) Code implementation
2. (0.5 pt) When Pacman believes that his death is unavoidable, he will try to end the game as soon as possible because of the constant penalty for living. Give an explanation as to why the Pacman rushes to the closest ghost in this case ?

3.3 Alpha-Beta Pruning (5 pts)

1. (5 pts) Code implementation

3.4 Expectimax (5.5 pts)

1. (5 pts) Code implementation
2. (0.5 pt) You should find that your ExpectimaxAgent wins about half the time, while your AlphaBetaAgent always loses. Explain why the behavior here differs from the minimax case.

3.5 Evaluation Function (6.5 pts)

1. (6 pts) Code implementation
2. (0.5 pt) What features did you use for your new evaluation function?

4 Self Analysis (3 pts)

1. What was the hardest part of the assignment for you?
2. What was the easiest part of the assignment for you?
3. What problem(s) helped further your understanding of the course material?
4. Did you feel any problems were tedious and not helpful to your understanding of the material?
5. What other feedback do you have about this homework?

5 Evaluation

Your code will be autograded for technical correctness. Please do not change the names of any provided functions or classes within the code, or you will wreak havoc on the autograder. If your code passes all the test cases in the autograder you will receive full points for the implementation.

However even if your code does not necessarily pass all the test cases, we will evaluate your code and then award you partial points accordingly. In such cases it would be even more beneficial if you could give a short description of what you tried and where you had failed and that would help us in giving you better points.

6 Submission Instructions

- For the final submission you will be turning in a zipped folder of the python files and a PDF document containing your responses to questions from previous sections.
- Please ensure all the submissions are done through canvas. Please do not email the instructor or the TA's with your submission. Submissions made via email will not be considered for grading.
- **Naming:** Your python files upload should be named in the format ⟨uid⟩-Proj⟨number⟩.zip where ⟨uid⟩ is your Utah uid and ⟨number⟩ is the Project number. **Ex:** u0006300-Proj2.zip
- For this project fill in portions of the files to edit. Once you have completed the code, zip your entire project folder, rename it as per the conventions stated above and submit it via canvas. Do not delete the other files present in the .zip file or change the names of any of those files in the project directory.
- **Written Answers:** Place all your written answers from Sections 3 and 4 in a single PDF document. This should be clearly named in the format ⟨uid⟩-Proj⟨number⟩-answers.pdf, where ⟨uid⟩ is your Utah uid and ⟨number⟩ is the Project number. Ex: u0006300-Proj0-answers.pdf Please make sure to write your name at the top of the document! Also, please upload this file separately from the .zip file.
- **Group Submissions:** If you haven't done this already for previous projects and if you are working in groups, You need to sign up to one of the project groups in the people page in canvas and under the groups tab. Please sign up to one of the "project groups" and ensure that your group member is signed up to the same group. This ensures that submissions from one of the group members counts for the entire group. Follow the file naming convention for the group member uploading the submission. i.e. do not mention the uid of your partner on the file name but mention your partner's details inside your answers.pdf . In the pdf, each group member is expected to provide their own responses to Section 4 "Self Analysis", but can work together on the code and the written responses to Section 3.

Note: Each group is limited to maximum 2 members, so if a group is taken already please choose the next free group.