Project 4: Ghostbusters

CS 6300 Artificial Intelligence

University of Utah Due: Apr 26, 2016 by 11:59:59 P.M

Project Material Courtesy : CS188 Berkeley course projects `http://ai.berkeley.edu`

In this project, you will design Pacman agents that use sensors to locate and eat invisible ghosts. You'll advance from locating single, stationary ghosts to hunting packs of multiple moving ghosts with ruthless efficiency.

As in Project 3, this project includes an autograder for you to grade your answers on your machine. This can be run 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 tracking.zip from here `http://ai.berkeley.edu/tracking.html` which will contain all the code and supporting files.

1 Files to edit

For this project you will need to edit the following files:

1. **bustersAgents.py** : Agents for playing the Ghostbusters variant of Pacman.
2. **inference.py** : Code for tracking ghosts over time using their sounds.

2 Ghostbusters and BNs(100 pts)

Your primary task in this project is to implement inference to track the ghosts. For the keyboard based game above, a crude form of inference was implemented for you by default: all squares in which a ghost could possibly be are shaded by the color of the ghost. Naturally, we want a better estimate of the ghost's position. Fortunately, Bayes' Nets provide us with powerful tools for making the most of the information we have. Throughout the rest of this project, you will implement algorithms for performing both exact and approximate inference using Bayes' Nets.

For all the problem titles described below, please refer to the link `http://ai.berkeley.edu/tracking.html` 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. Please ensure your code is readable and use comments in your code to make it more clear for the person reading your code.

2.1 Exact Inference Observation(12 pts)

1. (12 pts) Code Implementation

2.2 Exact Inference with Time Elapse(16 pts)

1. (16 pts) Code Implementation

2.3 Exact Inference Full Test (12 pts)

1. (12 pts) Code Implementation

2.4 Approximate Inference Observation (12 pts)

1. (12 pts) Code Implementation

2.5 Approximate Inference with Time Elapse(16 pts)

1. (16 pts) Code Implementation

2.6 Joint Particle Filter Observation(16 pts)

1. (16 pts) Code Implementation

2.7 Joint Particle Filter with Elapse Time(16 pts)

1. (16 pts) Code Implementation

3 Self Analysis

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?

4 Evaluation

Your code will be auto-graded 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 would receive full points for the implementation.

However even if your code does not necessarily pass all the test cases, we would 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.

5 Submission Instructions

- For the final submission you would be turning in a single zipped folder which should contain:
  - Folder having all the python files.(tracking folder in this case)
  - PDF document containing your responses to the questions in Section 3.
- For those of you who are doing it in teams, it is enough that one of the team members makes a submission. Please use the groups created in Canvas to upload your submission.
- Please ensure all the submissions are done through canvas. Please do not email the instructor or the TA with your submission. Submissions made via email would not be considered for grading.
- **Naming:**Your final upload should be named in the format $< uid >$-Proj4.zip where $< uid >$ is your Utah uid. **Ex:** u0006300-Proj4.zip
- For this project you should first unzip tracking.zip and then will fill in portions of code in the necessary files mentioned in Section 1. Do not delete any of the files or change the names of any of those files in the project directory.
- **Written Answers:** Place all written answers to questions in a single PDF document. This should be clearly named in the format $< uid >$-Proj4-answers.pdf, where $< uid >$ is your Utah uid. **Ex:** u0006300-Proj4-answers.pdf. Please make sure to write your name(s) at the top of the document!