

HW4 Project Report

1. Question 1 Exact Inference Observation:

- **Explanation:**

The goal is to update the beliefs of Pacman about the position of ghosts based on the observation between distance of itself and ghost. Provided the noisy distance of ghost's position we get the emission model which is probability of noisy distance given true distance. The calculation obeys the constraint that we have to consider ghost's position from list of valid and legal positions. Once a ghost has been eaten by a Pacman that need to put that ghost in a cell and thus handle the jail edge case so that Pacman again does not go for the same possibility. At the end of the complete process we get the new list of beliefs for Pacman with ghost's observed position.

- **Agent Result:**

Python Command	Result
python autograder.py -q q1 --no-graphics	<pre>Question q1 ===== *** q1) Exact inference stationary pacman observe test: 0 inference errors. *** PASS: test_cases/q1/1- ExactObserve.test *** q1) Exact inference stationary pacman observe test: 0 inference errors. *** PASS: test_cases/q1/2- ExactObserve.test *** q1) Exact inference stationary pacman observe test: 0 inference errors. *** PASS: test_cases/q1/3- ExactObserve.test *** q1) Exact inference stationary pacman observe test: 0 inference errors. *** PASS: test_cases/q1/4- ExactObserve.test ### Question q1: 3/3 ### Finished at 3:27:12 Provisional grades ===== Question q1: 3/3 ----- Total: 3/3</pre>

2. Question 2 Exact Inference with Time Elapse:

- **Explanation:**

The goal is to update the Pacman's belief about the ghost's position using the time step passing from the current state. First, we will calculate the new position distribution of the ghost's position using the previous position as well as the Pacman's current position. Here also we will obey the constraint of using only legal and valid action for the ghost's movement. For each new position for the newly calculated distribution we will find the probability in accordance with the old belief and finally we will get a list of new positions for the ghosts. Once completed we will update the old belief of Pacman with the new one.

- **Agent Result:**

Python Command	Result
python autograder.py -t test_cases/q2/2-ExactElapse --no-graphics	*** None) Exact inference elapseTime test: 0 inference errors. *** PASS: test_cases/q2/2-ExactElapse.test
python autograder.py -q q2 --no-graphics	Question q2 ===== *** q2) Exact inference elapseTime test: 0 inference errors. *** PASS: test_cases/q2/1-ExactElapse.test *** q2) Exact inference elapseTime test: 0 inference errors. *** PASS: test_cases/q2/2-ExactElapse.test *** q2) Exact inference elapseTime test: 0 inference errors. *** PASS: test_cases/q2/3-ExactElapse.test *** q2) Exact inference elapseTime test: 0 inference errors. *** PASS: test_cases/q2/4-ExactElapse.test ### Question q2: 4/4 ### Finished at 3:49:46 Provisional grades ===== Question q2: 4/4 ----- Total: 4/4

3. Question 3 Exact Inference Full Test:

- **Explanation:**

The goal is to build a greedy buster agent that will charge the closest ghosts. And we will find the correct and best action for Pacman so that it will work like greedy agent. First, we will get all the living ghosts in the current distribution and validate them using the actual belief of Pacman. Then for each living ghost we will find out the most likely position which are not yet captured. And then for each action in the legal and allowed list of action we will calculate the maze distance between Pacman and ghost's position. We will choose the minimum distance and return the best action that used for the calculation.

- **Agent Result:**

Python Command	Result
python autograder.py -t test_cases/q3/3-gameScoreTest --no-graphics	[Distancer]: Switching to maze distances Average Score: 764.5 Scores: 754, 765, 771, 767, 771, 764, 775, 760, 753, 765 Win Rate: 10/10 (1.00) Record: Win, Win, Win, Win, Win, Win, Win, Win, Win, Win *** Won 10 out of 10 games. Average score: 764.500000 *** *** smallHunt) Games won on None with score above 700: 10/10 *** PASS: test_cases/q3/3-gameScoreTest.test
python autograder.py -q q3 --no-graphics	Question q3 ===== *** q3) Exact inference full test: 0 inference errors. *** PASS: test_cases/q3/1-ExactObserveElastic.test *** q3) Exact inference full test: 0 inference errors. *** PASS: test_cases/q3/2-ExactObserveElastic.test [Distancer]: Switching to maze distances Average Score: 755.7 Scores: 739, 757, 749, 749, 755, 784, 777, 735, 745, 767 Win Rate: 10/10 (1.00) Record: Win, Win, Win, Win, Win, Win, Win, Win, Win, Win *** Won 10 out of 10 games. Average score: 755.700000 ***

	<p>*** smallHunt) Games won on q3 with score above 700: 10/10</p> <p>*** PASS: test_cases/q3/3-gameScoreTest.test</p> <p>### Question q3: 3/3 ###</p> <p>Finished at 4:05:01</p> <p>Provisional grades</p> <p>=====</p> <p>Question q3: 3/3</p> <p>-----</p> <p>Total: 3/3</p>
--	--

Autograder Report:

python autograder.py

Question q1

=====

*** q1) Exact inference stationary pacman observe test: 0 inference errors.
 *** PASS: test_cases/q1/1-ExactObserve.test
 *** q1) Exact inference stationary pacman observe test: 0 inference errors.
 *** PASS: test_cases/q1/2-ExactObserve.test
 *** q1) Exact inference stationary pacman observe test: 0 inference errors.
 *** PASS: test_cases/q1/3-ExactObserve.test
 *** q1) Exact inference stationary pacman observe test: 0 inference errors.
 *** PASS: test_cases/q1/4-ExactObserve.test

Question q1: 3/3

Question q2

=====

*** q2) Exact inference elapsedTime test: 0 inference errors.
 *** PASS: test_cases/q2/1-ExactElapse.test
 *** q2) Exact inference elapsedTime test: 0 inference errors.
 *** PASS: test_cases/q2/2-ExactElapse.test
 *** q2) Exact inference elapsedTime test: 0 inference errors.
 *** PASS: test_cases/q2/3-ExactElapse.test
 *** q2) Exact inference elapsedTime test: 0 inference errors.
 *** PASS: test_cases/q2/4-ExactElapse.test

Question q2: 4/4

Question q3

=====

```
*** q3) Exact inference full test: 0 inference errors.
*** PASS: test_cases/q3/1-ExactObserveElapse.test
*** q3) Exact inference full test: 0 inference errors.
*** PASS: test_cases/q3/2-ExactObserveElapse.test
[Distancer]: Switching to maze distances
Average Score: 755.7
Scores:      739, 757, 749, 749, 755, 784, 777, 735, 745, 767
Win Rate:    10/10 (1.00)
Record:      Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
*** Won 10 out of 10 games. Average score: 755.700000 ***
*** smallHunt) Games won on q3 with score above 700: 10/10
*** PASS: test_cases/q3/3-gameScoreTest.test
```

Question q3: 3/3

Question q4

=====

```
*** Method not implemented: getBeliefDistribution at line 348 of inference.py
*** FAIL: Terminated with a string exception.
```

Question q4: 0/3

Question q5

=====

```
*** Method not implemented: getBeliefDistribution at line 348 of inference.py
*** FAIL: Terminated with a string exception.
```

Question q5: 0/4

Question q6

=====

```
*** Method not implemented: getBeliefDistribution at line 535 of inference.py
*** FAIL: Terminated with a string exception.
```

Question q6: 0/4

Question q7

=====

```
*** Method not implemented: getBeliefDistribution at line 535 of inference.py
*** FAIL: Terminated with a string exception.
```

Question q7: 0/4

Finished at 4:06:40

Provisional grades

=====

Question q1: 3/3

Question q2: 4/4

Question q3: 3/3

Question q4: 0/3

Question q5: 0/4

Question q6: 0/4

Question q7: 0/4

Total: 10/25

Your grades are NOT yet registered. To register your grades, make sure to follow your instructor's guidelines to receive credit on your project.