p4_tracking_q1_exact_inference_observation

Question 1 (3 points): Exact Inference Observation

In this question, you will update the `observe` method in `ExactInference` class of `inference.py` to correctly update the agent's belief distribution over ghost positions given an observation from Pacman's sensors. A correct implementation should also handle one special case: when a ghost is eaten, you should place that ghost in its prison cell, as described in the comments of `observe`.

To run the autograder for this question and visualize the output:

```
python autograder.py -q q1
```

As you watch the test cases, be sure that you understand how the squares converge to their final coloring. In test cases where is Pacman boxed in (which is to say, he is unable to change his observation point), why does Pacman sometimes have trouble finding the exact location of the ghost?

Note: your busters agents have a separate inference module for each ghost they are tracking. That's why if you print an observation inside the `observe` function, you'll only see a single number even though there may be multiple ghosts on the board.

Hints:

- You are implementing the online belief update for observing new evidence. Before any readings, Pacman believes the ghost could be anywhere: a uniform prior (see `initializeUniformly`). After receiving a reading, the `observe` function is called, which must update the belief at every position.

- Before typing any code, write down the equation of the inference problem you are trying to solve.
- Try printing `noisyDistance`, `emissionModel`, and `PacmanPosition` (in the `observe` function) to get started.
- In the Pacman display, high posterior beliefs are represented by bright colors, while low beliefs are represented by dim colors. You should start with a large cloud of belief that shrinks over time as more evidence accumulates.
- Beliefs are stored as `util.Counter` objects (like dictionaries) in a field called `self.beliefs`, which you should update.
- You should not need to store any evidence. The only thing you need to store in `ExactInference` is `self.beliefs`.