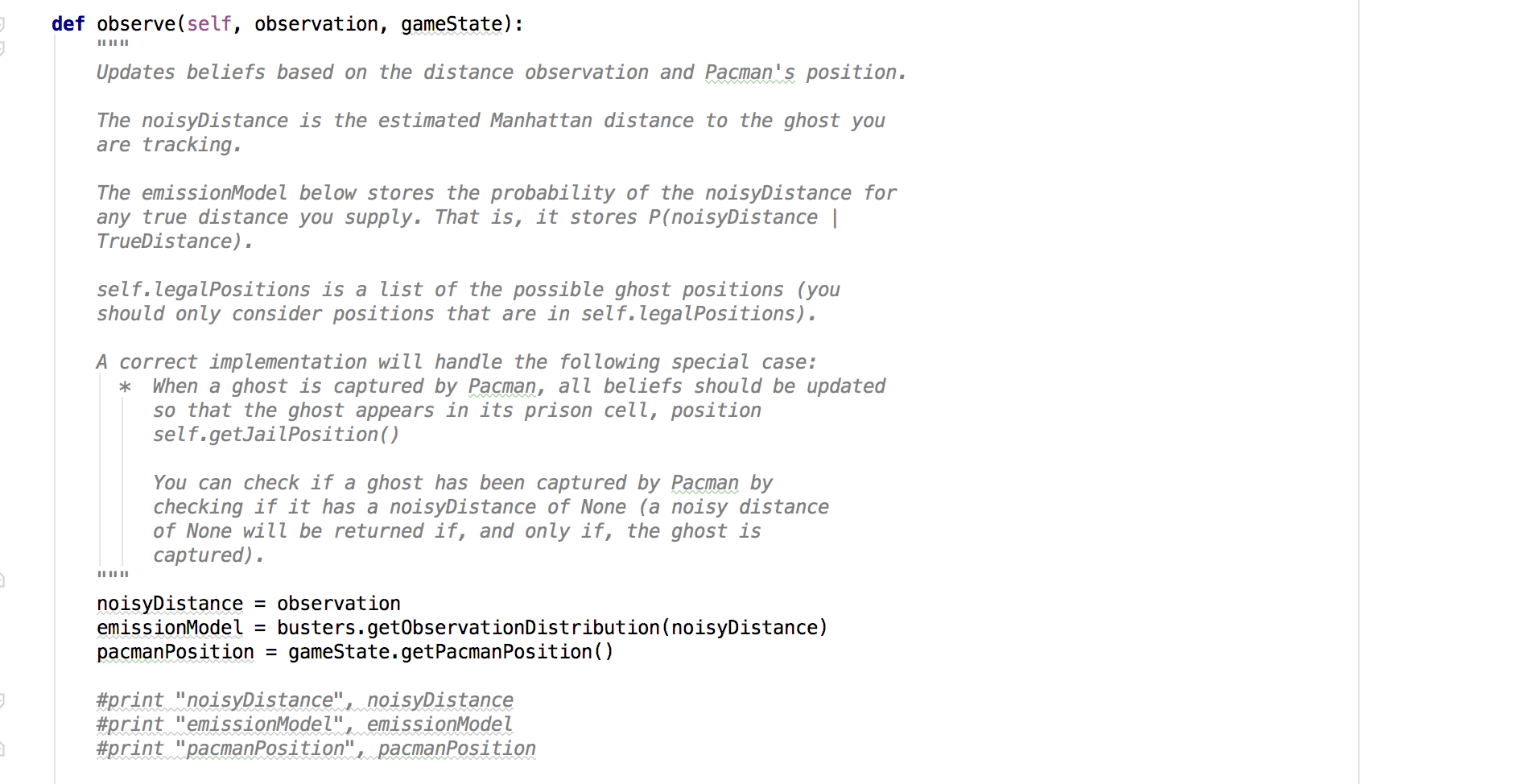
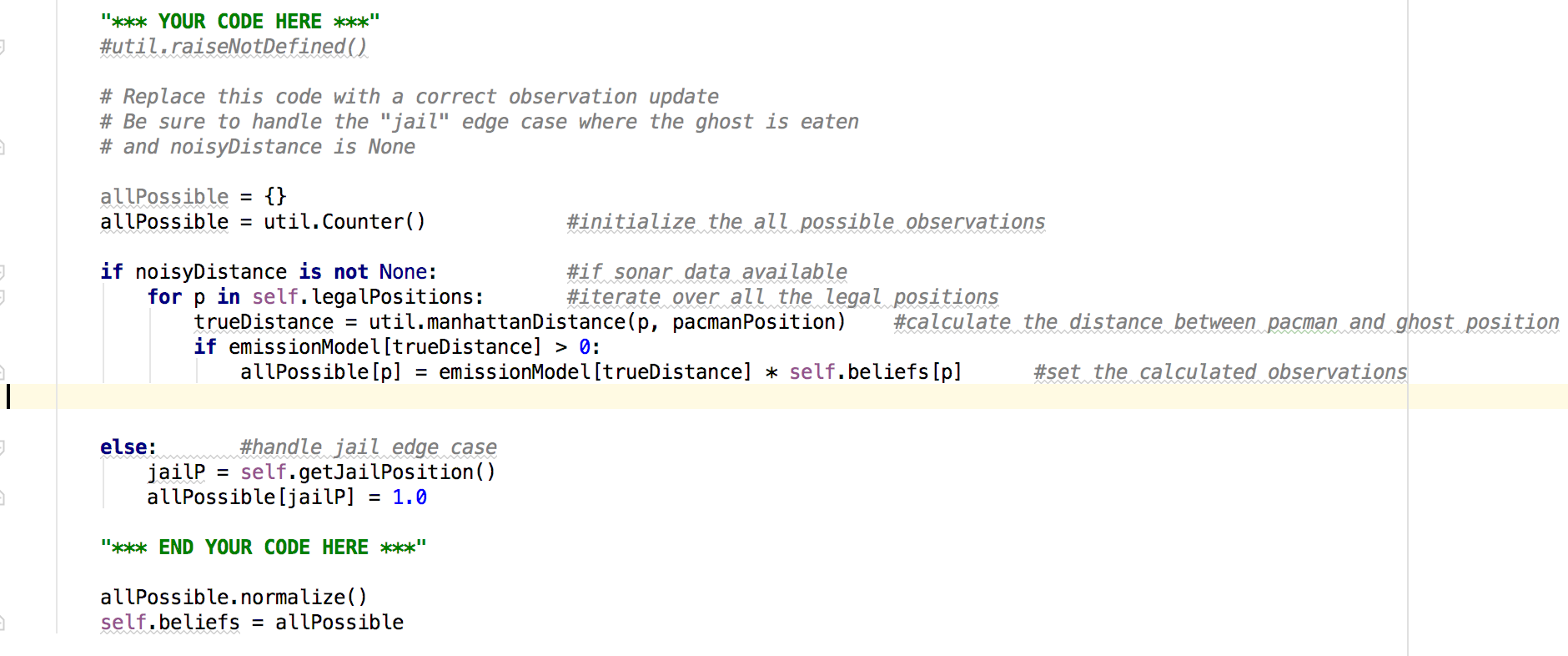
**HW4 Project Details Document**

1. ***Question 1***Exact Inference Observation:

* **Code Details:**

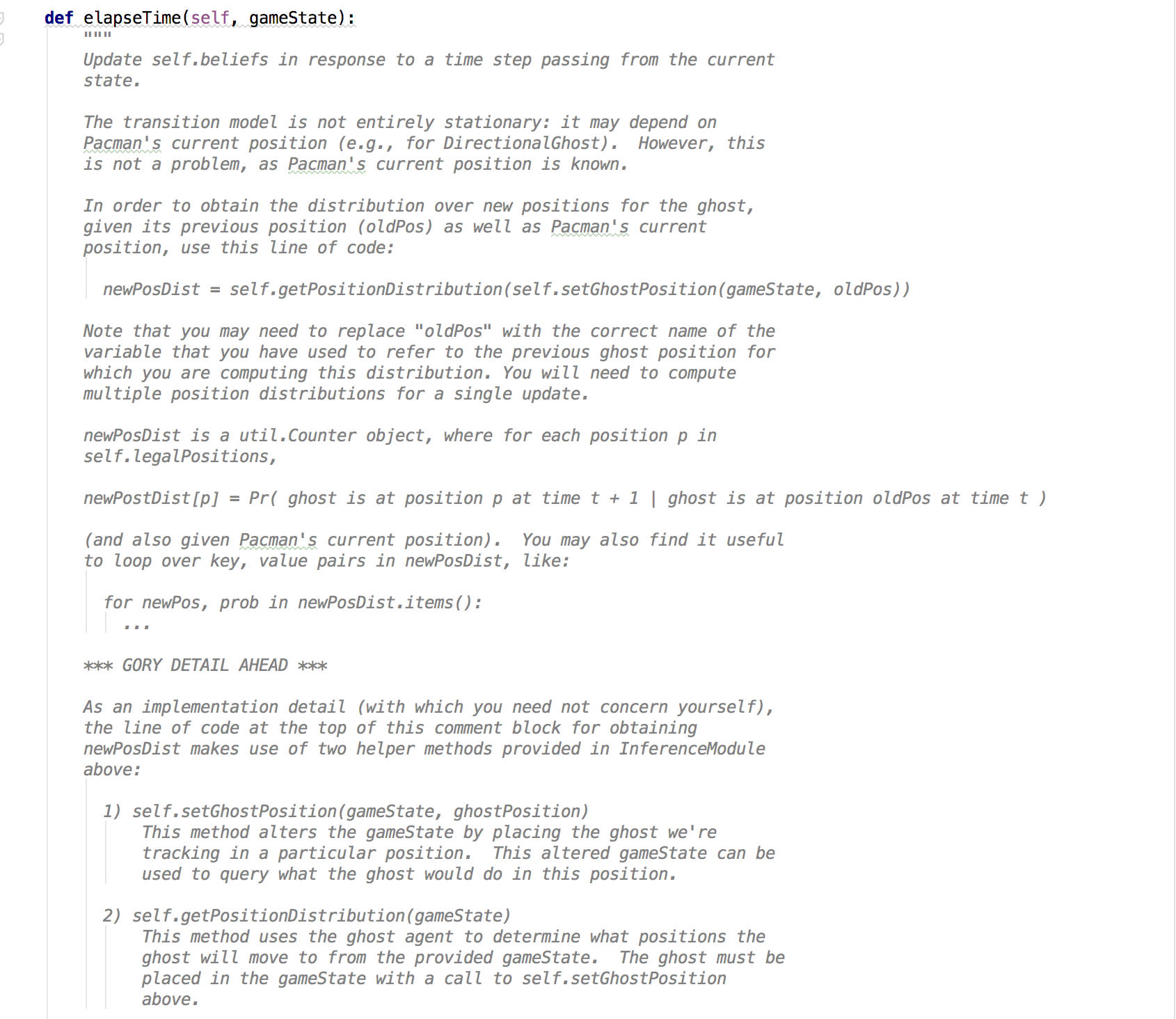


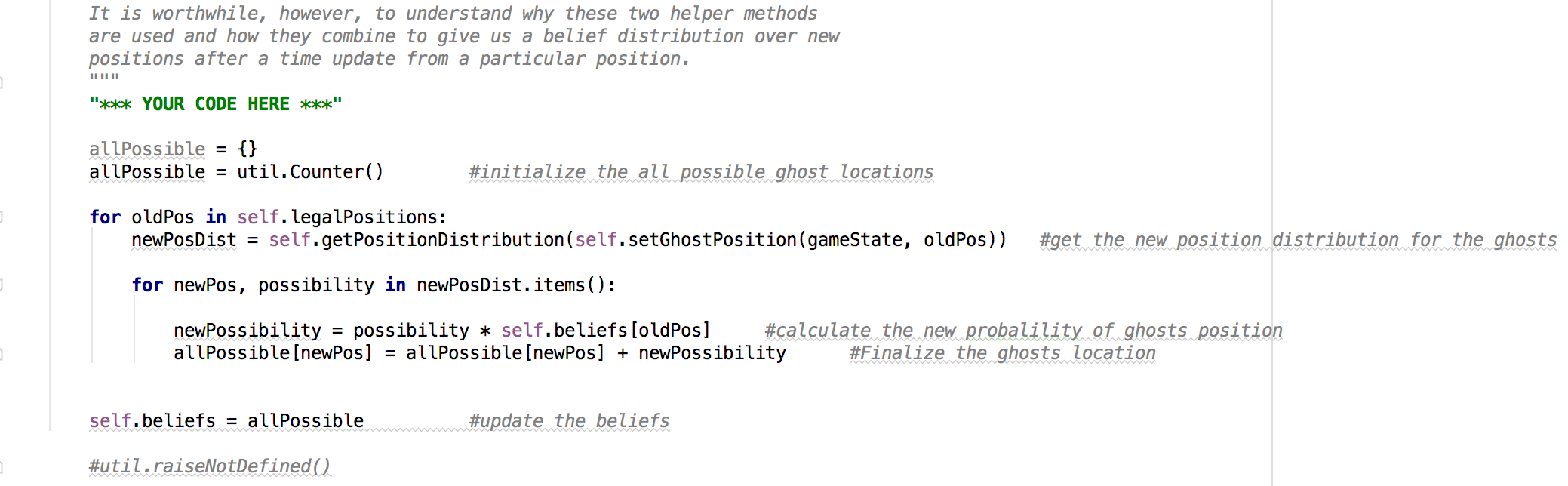


* I have updated the code in for observer() method inside ExactInference class in inference.py file.
* We were provided with noisy distance, emission model and Pacman position to calculate the observation.
* Get all the existing Pacman beliefs in allPossible dictionary.
* If noisyDistance is available then:
  + Loop over the legal positions:
    - Calculate the Manhattan distance between Pacman’s position and ghost’s position
    - If any positive value exist for emission model for the current distance then:
      * Calculate the new observation using the emission model and old Pacman’s belief.
* Handle the jail edge case if noisy distance not available. This means that the ghost is already eaten by Pacman.
  + Get the position of the eaten ghost
  + Mark that in the Pacman’s belief system.
* Normalize the calculated observations
* Update the old belief system with the new one for Pacman.

1. **Question 2**Exact Inference with Time Elapse:

* **Code Details:**

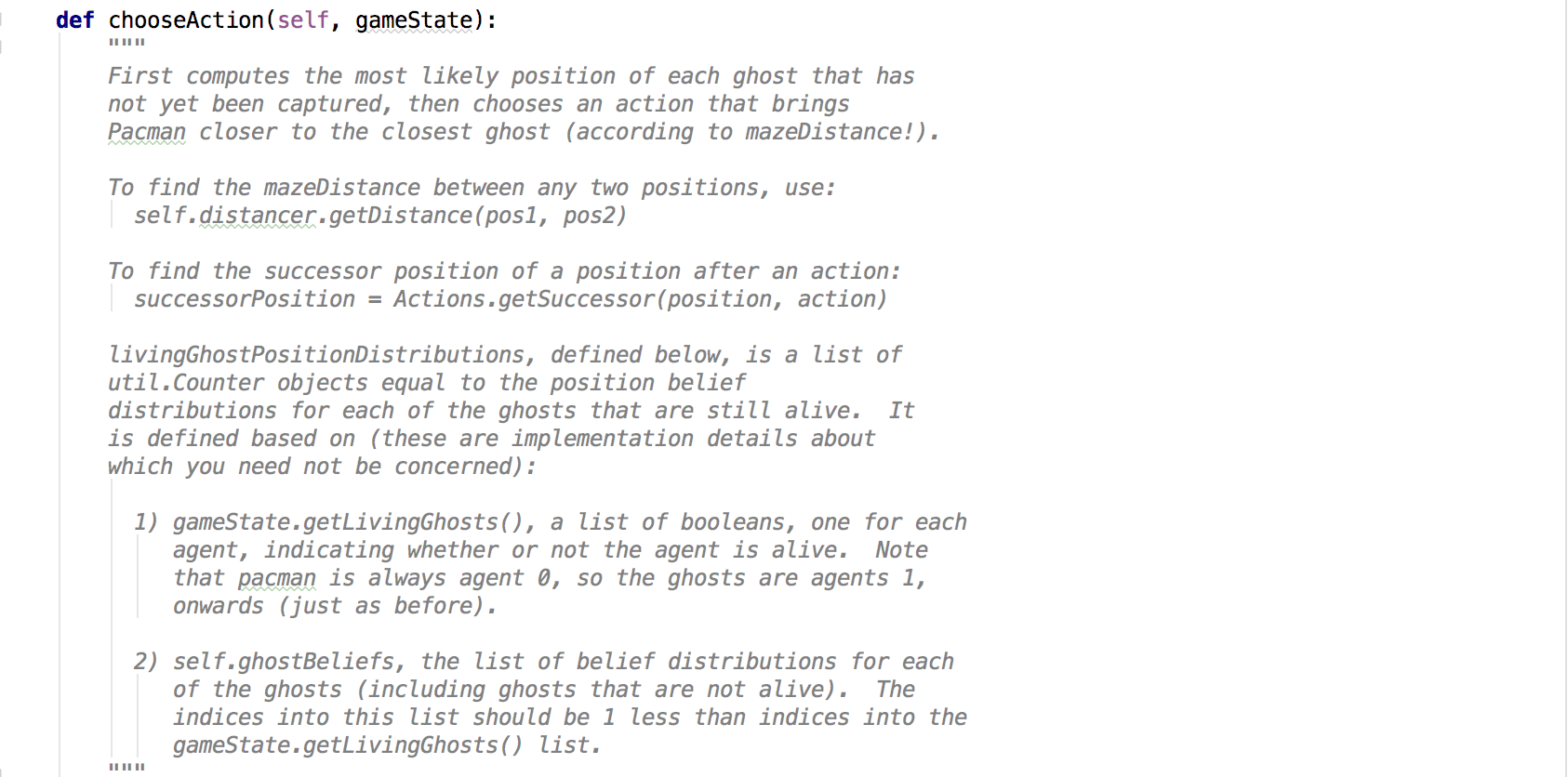


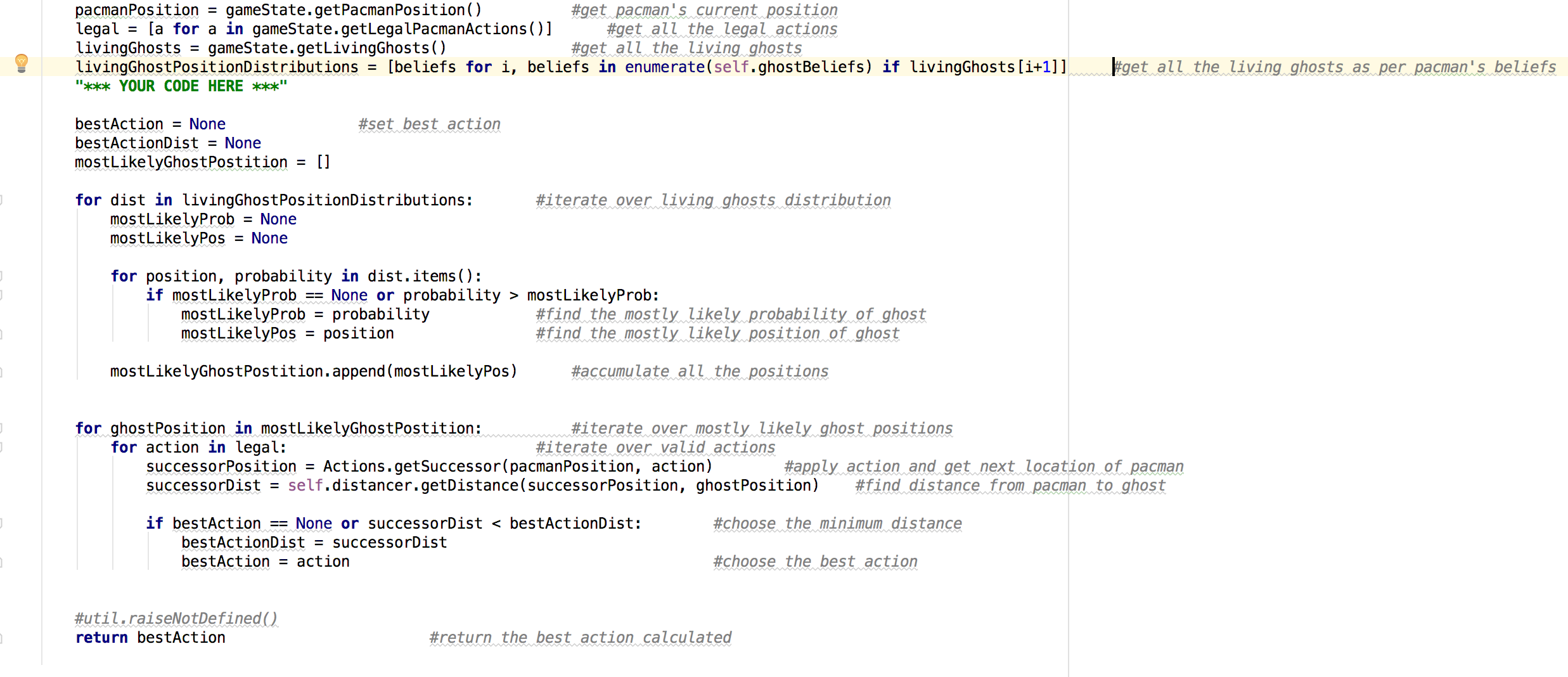


* I have updated the code in for elapseTime() method inside ExactInference class in inference.py file.
* Get all the beliefs of Pacman in allPossible dictionary.
* Loop over all the positions in legal positions:
  + Get the new position distribution for ghosts
  + Loop over new positions and probabilities for this new distribution:
    - Calculate the new possibility using the old belief system and new probability value
    - Finalize the new position for the ghosts
* Update the belief system to new values for Pacman.

1. **Question 3** Exact Inference Full Test:

* **Code Details:**





* I have updated chooseAction() method in GreedyBusterAgent class in bustersAgents.py
* We were provided with Pacman’s current position in the game.
* Get all the legal actions that are allowed for Pacman.
* Get all the living ghost’s positions and validate them as per the Pacman’s actual belief system.
* Define bestAction, bestActionDist and mostLikelyGhost positions.
* Loop over the distribution of living ghosts:
  + Define mostlikelyprob and mostlikelypos
  + For each position and probability over each distribution:
    - If mostlikelyprob is none or less than the distribution probability then:
      * Set this as new mostlikely prob
      * Set mostlikelypos as distribution position
  + Append the new position to the ghost position list
* For each ghost positions in ghost position list:
  + For each legal actions:
    - Get the successor position for Pacman’s current position
    - Find the distance from Pacman’s successor to ghost
    - If best action is none or successor’s distance is less than best action distance then:
      * Set this as new best distance action
      * Set this action as new best action
* At the end of the execution return the best action from this greedy buster agent.