Reinforcement Learning Submission

QUESTION: Observe what you see with the agent's behavior as it takes random actions. Does the smartcab eventually make it to the destination? Are there any other interesting observations to note?

The smart cab, given infinite time, would eventually make it to the goal. But in finite time, since it is taking psuedo-random movements, there is no reason to expect that it would make it to the goal. Interesting behavior that the cab exhibited was that it waited for the lights (or gates in the environment).

**QUESTION:** What states have you identified that are appropriate for modeling the smartcab and environment? Why do you believe each of these states to be appropriate for this problem?

The states given to the agent as inputs are the status of the lights at the intersection, the oncoming traffic status, the right traffic status and the left traffic status. The input is given in dictionary format as shown in the following example: {'light': 'red', 'oncoming': None, 'right': None, 'left': None}. I have chosen to use the ‘light’, ‘oncoming’, and ‘left’ inputs as states to model the smartcab. In addition, I have chosen to use the ‘next\_waypoint’ property of the agent as a ‘direction’ state to model the smartcab.

The reason that the ‘light’ property is important is that the Agent needs to know the status of the light in front of it in order to make a decision of what to do next. If the light is red, it might be more optimal to go right instead of waiting for the light to turn green. If the agent were to break this rule, it would result in a large negative reward value.

The reason the ‘left’ property (I.e presence of traffic from the left) is important is because if the light is red and there is traffic present on the left, the car must wait for the left present car to pass through before it can safely make a right turn.

The reason the “oncoming” traffic is important is because if our smartcab needs to make a left turn on a green light, it must wait for oncoming traffic to pass before it can safely do so.

Finally, the ‘direction’ state is needed so our smartcab knows which way to go. Different directions should be associated with different awards

OPTIONAL: How many states in total exist for the smartcab in this environment? Does this number seem reasonable given that the goal of Q-Learning is to learn and make informed decisions about each state? Why or why not?