Matthew Ignal
Report for P4

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 agent moves randomly around the grid, occasionally making it to the destination (about 22%), but more frequently running out of time.

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 that are appropriate for modeling the smartcab and its environment are the lights, traffic (left, right, and oncoming), and the next waypoint. It's obviously important for the agent to behave appropriately based on the lights and traffic, and it needs to know where to go as well. Deadline was not included, as the agent receives greater rewards for traveling in the direction of the next waypoint, so it is unnecessary. As we are prioritizing safety, an agent running low on time should not violate traffic rules to reach the destination, as including a deadline state might incur. These were stored as a tuple, which is hashable and can therefore act as a dictionary key in the Q-learning table.

QUESTION: What changes do you notice in the agent's behavior when compared to the basic driving agent when random actions were always taken? Why is this behavior occurring?

The main changes in the agent's behavior following the implementation of the basic Q-learning table was that traffic rules were followed as the agent tried to make its way to the destination. This behavior is occurring because the Q-learning table is storing the rewards for each action in a given state and is then choosing the best action based off these rewards.

QUESTION: Report the different values for the parameters tuned in your basic implementation of Q-Learning. For which set of parameters does the agent perform best? How well does the final driving agent perform?

A few changes were made to a basic Q-learning algorithm. First, if the state had not been encountered, I wanted the agent to pick an action at random. If the state had been encountered, I wanted the agent to pick more far more actions at random towards the beginning and far fewer later (epsilon = 1/trial #). For testing, I didn't want to incur the risk of taking *any* actions at random if the state had been encountered. If there was a tie according to the Q-learning algorithm, I wanted the agent to take None if it was available, as we are prioritizing safety.

For 100 test trials (after 100 initial trials) the following performance metrics for tuned parameters alpha and gamma were:

Alpha	Gamma	Destination Reached Rate	Illegal Actions per Trial
1.0	1.0	0.0	0.0
1.0	0.5	0.95	0.02
0.5	0.2	0.99	0.0
0.5	1.0	0.0	0.0
1.0	0.2	0.93	0.0
0.8	0.2	0.95	0.0
0.5	0.5	0.96	0.0
0.2	0.2	1.0	0.0
0.5	0.0	0.94	0.0

Using an alpha of 0.2 and a gamma of 0.2, I was able to get 100 successful trips out of 100 with the smartcab with no illegal actions taken.

QUESTION: Does your agent get close to finding an optimal policy, i.e. reach the destination in the minimum possible time, and not incur any penalties? How would you describe an optimal policy for this problem?

The agent learns to not perform illegal actions and move in the direction of the destination. After just a few dozen trials, the smartcab seems to avoid incurring penalties while reaching the destination in the minimum possible time. An optimal policy would have the smartcab driving legally and reaching its destination in the least possible time while never performing illegal moves, so the performance metric of illegal actions was included. With a low gamma and a medium-low alpha, both of these metrics for an optimal policy are satisfied.