***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 behaves randomly as one would expect. Eventually, it does reach the destination, although, in theory I supposed it could never reach it. The agent does not take illegal actions, but instead chooses them and remains in place with a penalty.

***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?*

My environment state is a tuple with four pieces of information: the color of the light; whether there is either no oncoming traffic or the oncoming traffic is turning left; whether or not there is traffic on the left going forward; and what the next waypoint is. The agent has four possible actions in all environment states. The environment state provides all the information necessary to know whether any given move is legal and which direction the smartcab should go in next, which is all the information necessary for the smartcab to navigate legally to the destination.

***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?*

There are 2*2*2*3 = 24 possible environment states in this model and 4 legal actions giving a total of 96 states for the smartcab. Whether or not this number is reasonable depends upon how quickly you want the smartcab to learn to operate in this environment and how often each environment state will be encountered. Although the total number of states is very low, the scarcity of other cars, it could take a relatively long time for the smartcab to encounter every state-action pair. However, because optimal action for any given state is always the same, even given the 100 trial limit the agent should have plenty of time to learn and make informed decisions about all of them.

***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?*

At first the agent's behavior does not seem that different from the random agent, but very quickly it starts to go directly from the starting point to the destination with few to no detours and increasingly fewer illegal moves. The agent is actually mostly exploring unseen states in the beginning, and then transitioning to mostly exploiting the predicted utilities of each state. This is occurring because I initialized the Q-Table with high values, which promotes exploration of unseen states. However,

because the alpha I use is high, the table quickly approximates the true values, and the agent switches to exploitation.

***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?*

I tried alpha values from 0.1 to 1.0 and gamma values from 0 to 0.8. I tried decaying the alpha value over time as well as not. The lower the alpha value, the longer the policy takes to converge, and if it is decayed too quickly, the agent may end up stuck with a non-optimal policy. The policy also seems to reach the optimum faster for lower values of gamma. The method by which I implemented epsilon work by adding a random scaled value to each of the expected values and then taking choosing the action corresponding to the new max. This allows the random actions to take into account the learned values and in the end provides the same result as exploitation would, in this case. I fixed epsilon at 0.1, however due to the optimistic starting conditions and this method of implementation of epsilon, any using any value for epsilon works approximately as well. In fact, using sufficiently optimistic starting conditions is sufficient My agent performs best (optimally when all states have been explored) with alpha set to 1 and gamma set to 0.

***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 optimal policy for this problem is the policy that results in the shortest route with regard to time (not number of actions) and does not attempt any illegal actions (as they are always inferior to the "None" action). In practice, this means only making moves that go to the next waypoint when they are legal and choosing the "None" action if they are not. Once my agent has encountered every state-action pair, it operates according the the optimal policy.