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?*

Answer: The agent behavior appears to be completely random in this case. The agent is roaming in the environment without any sense of direction. Agent is accumulating negative reward because of breaking traffic rules and for causing accidents. The agent is also missing deadlines because of that.

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?*

Answer: I have identified the following states, to track the state of a smart car:-

a) "Okay to go left" (state=0): The agent will enter into this state, if the next way point is "left" and inputs from the all the directions: - "left", "right" and "oncoming" is None. If no other car is coming in from any of these three directions, it is safe to take a left turn. So, in this case the cab will enter into state 0.

b) "Okay to go forward" (state=1): The agent will enter into this state, if its next waypoint is forward and no other cab is coming in either from the left or from the right direction.

c) "Okay to go right" (state=2): The agent will enter into this state, if its next waypoint is right. Since, right is mostly a free turn in US roads. The agent will be allowed to make a turn without any other checks and will enter into state 2.

d) "Collision state" (state=3): If the agent doesn't enter into any of the three states, then the agent will enter into state=3. This is a collision state.

e) "Stop at red signal" (state=4): If the light is red, then the can will enter into the red signal state. This is state=4, or the "red signal" state.

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?*

Answer: If we take the rough count values, the total number of states, which we will need to model this world, will be 48. This number doesn't seem reasonable since in order to learn a valid Q function by Q-learning we need to visit every state action pair almost infinite number of times, to model the true value of the Q function. In addition, in the real world we won't model all the intersection in a town as a state of the smart car.

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?*

Answer: After many iterations of the Q-learning algorithm, the agent seems to be learning the approximation of the optimal policy. The agent seem to reaching the goal state, more rapidly compared to way it was roaming in the environment at random. Also, the agent is not missing the deadlines, as it was doing before. Here's is the q_matrix from the algorithm, after 51 trails of the algorithm.

```
Action [ None, forward, left, right]
State 0 [[1 1 2 2]
State 1  [0 3 2 2]
State 2  [1 1 1 2]
State 3 [1 2 2 2]
State 4 [0 0 0 0]]
```

This behavior is occurring because the agent is learning the approximation of the true Q(s, a) function(or the policy) in this environment.

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?*

Answer: I trained the smartcab with the two choices of the learning rate parameter in the q_learning algorithm.

a) learing_rate = 0.1 – While using learning rate of 0.1, the smartcab took more time to reach to a good approximation of the optimal policy. After around 51 or 52 iterations, the smartcab was able to traverse optimally in the environment.

b) Learning_rate = 0.5 – While using the learning rate of 0.5, the smartcan took less time to reach a good approximation of the optimal policy. It started performing better actions, after only 15 or 16 trails in the experiment.

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?*

Answer : Yes, the agent is able to reach the goal in most cases. After, having a good approximation of the q_matrix. The agent is able to traverse in the environment in a systematic manner. After the initial, training the agent no longer misses the deadline. The optimal policy for this environment will be take the route in the environment which gives the maximum reward in long term.