**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 cab moves around randomly as expected, occasionally reaching the destination but more often not. It often attempts to go forward or left when at a red light and thus stays in place. The borders are not hard borders but rather transports the agent to the opposite edge.

**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 state takes light, left, oncoming from inputs as well as next\_waypoint. These are appropriate because it supplies the information about what the smartcab can do at the intersection, and where the smartcab wants to go next. Right can be ignored because it has no relevance to what action the smartcab can take (assuming that it obeys traffic laws).

**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?

How many total states can be determined by looking at each state variable:

* Light: Green, Red 🡪2
* Left: forward, right, left, none 🡪4
* Oncoming: forward, right, left, none 🡪4
* Next\_waypoint: forward, right, left 🡪3

Total States = 96. This seems like a reasonable number of states as we attempt to run with a large number of trials, especially given the fact that there is limited traffic so many of the states with traffic from left or oncoming will not be reached

**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 agent’s behavior quickly becomes much better than when the random action is always taken because it is learning its policy at each step (populating the qtable) and then acting on what it has learned (using the policy).

**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?

Alpha of .5, gamma of .2 and epsilon of .1. The agent performs well with those values, and most values as long as alpha is >0 and epsilon is low. The final agent performs very well, reaching the desitination on all trials after the first few.

**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?

Yes, the agent performs well, reaching the destination without any penalties within the allotted time. The optimal policy for this problem is essentially to go to the next waypoint if traffic allows, and to stay in place if traffic doesn’t