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?*

Since the deadline check is set to false, the learning agent navigates forever within one single trial until it reaches the destination without any training algorithm being involved. I do not notice other interesting observations at this stage.

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?*

traffic light combinations (green, red, in total 2), is time running out (yes, no, in total 2), does our car crash on another car (yes, no, in total 2), position (x, y, grid size 6*8 = 48)

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?*

Based on my description above, the total amount of states = 2 * 2 * 2 * 48 = 284

I do not have answer for the other parts for now

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?*

Our test car in color red stucks at the same location within same trial very often, I do not know if my implementation of QLearning is incorrect or it is because it reaches a local optimal, there staying in the same location is considered as the best act at specific state; and that is the main reason that we have consecutive stochastic trials to reach a global optimal

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?*

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?*