1. When the car is allowed to take random actions and the 'enforce deadline' parameter is set to False, the car is ultimately able to reach the destination but takes a rather long time.

2. State includes the inputs and the location of the next waypoint.

State Variables:

1. Light

2. Ongoing

3. Left

4. Right

5. Next Waypoint

The deadline is not important variable to decide the state of the red car. Seeing all such states (as defined above) will help the Q-learning algorithm to visit every possible condition the car can see (after a quite long training time)

3. Why these variables in the state definition?

* The inputs are important as the car would receive very negative rewards if it is breaking any traffic rules. Thus, the Q-learning algorithm must see the variables such as Light, Ongoing, Left, Right. Breaking rules will not be good for our agent. Thus, it is important that these variables are included in the state definition.
* The next\_waypoint tells us the next waypoint to visit. Also the agent should act optimally such that it reaches the destination without breaking rules and taking the shortest path. Thus, inclusion of this variable in the state definition is important.
* The deadline variable should not be included in the state definition because the deadline will not tell us anything specific and will only need to a new state each time. This is not good as our agent would not be able to learn anything.

4. By application of Q-learning, the agent will be able to reach the destination without breaking rules and making use of the shortest path. The point is that the actions are firstly chosen randomly. Now, the max(a) Q(s, a ) gives us the utility of the state we are in. Thus, after the agent has seen all actions related to a state the agent can then choose the best action. That would be moving closer to the optimal policy. The agent gets bigger reward for reaching the destination, thus this strengthens the optimal policy more. The presence of the next\_waypoint in the state helps the agent to take the best action trying to take the shortest path and avoiding any accidents. Any illegal action will make the agent get negative rewards. Thus the policy will be changing (if necessary, that is if the Q-values changes) and getting closer to the optimal policy.

5. The following alpha and gamma values were tried for the algorithm:

* gamma = 0.3,alpha = 0.4

The no of examples that were obtained right: 98/100. The agent learns quite fastly and chooses the route that may not be the shortest but the one that is the least accident causing. Thus, it chooses the path which can minimize breaking of traffic rules over the shortest path to the next\_waypoint. The no of examples which got right in an iteration also would vary by a few numbers( +2, -2) since the random initialization may get sometimes lucky for the agent. However, it ultimately reaches the near optimal policy and is able to choose the best action for the state it is present in.

* 1, 1

The no of examples that were obtained right: 67/100. The agent is not able to learn very well this time. This is because there are no discounted weights. The car sometimes chooses the same action again and again and getting stuck.

Thus 0.3, 0.4 seems to be a good choice for gamma and alpha respectively. The agent learns quite fastly and chooses the route that may not be the shortest but the one that is the least accident causing.