1. In your report, mention what you see in the agent's behavior. Does it eventually make it to the target location?

Agent moves randomly. It can reach the target location accidentally but a success rate is very low even when deadline check is disabled.

2. Justify why you picked these set of states, and how they model the agent and its environment.

My set of states state consists of:

- next waypoint
- traffic light
- oncoming traffic
- traffic from the left

The next waypoint is the most important state as an agent is rewarded every time it follows waypoints. Also following waypoint agent can actually reach the target location.

The rest of states (light, oncoming and left) are important to follow traffic rules and not to be fined.

3. What changes do you notice in the agent's behavior?

Agent starts with arbitrary moves but later becomes more accurate and can reach the target location most of the time. Also, it gets penalties less often.

4. Report what changes you made to your basic implementation of Q-Learning to achieve the final version of the agent. How well does it perform?

I tried different values for initial Q, learning rate and discount factor. Good result were achieved with following values:

- initial Q = .5
- learning rate = .1
- discount factor = .3

Now the agent can get to the destination with no penalties.

But as the environment is completely deterministic a learning rate can be set to 1, and a discount factor can be set to 0. With low initial Q it gives the best results.

5. 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?

The resulted policy allows to reach the destination with almost no penalties but the time is not minimum possible. It is expected as the agent are not rewarded for a smaller time.