

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

When agent given unlimited time to reach destination:

- **mention what you see in the agents behavior**
  - The agent has no state, no action, and reward of 0.0 at every time step.
- **does he make it to target location**
  - Initially there is no q training being implemented with the agent. Nor any other forms of getting to the target location. Overall since the agent has no action, it remains still and only 'reaches' target destination, if it is randomly selected to start there.

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

- Next location
- intersection states
- Although it is important for the current deadline to be met, it isn't a valuable addition to the state because the agent should attempt to find the fastest route every single time. By understanding the destination coordinate, next location, and intersection state, the agent should develop a policy to arrive at the destination while avoiding any slower/longer paths.
- They model the agents environments since it uses the state set in order to meet the objective of reaching the destination before the deadline.
- Is it possible to add the destination into the state?

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

I see no changes yet. I believe adding the destination will change the agent's behavior. Although what is the variable for destination? Also I'm curious on how `self.next_waypoint` works

**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?**

Once I get the previous question down, I would improve my algorithm using the epsilon greedy algorithm.

**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?**

What are the parameters and what would they point towards for me to know I implemented the right q learning algorithm?