

Name: David Chan Chi Fung
Project: Udacity Training a Smartcab To Drive

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

The agent simply walks around very randomly, without eventually making it to the destination. I believe there is a probability it might make it to the destination, but this is will happen very rarely and occasionally.

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

I picked the states **light**, **left**, **right** and **next_waypoint**.

I picked up the **light**, **left** and **right** states so that the agent can properly model the traffic in its environment.

And finally, I picked the **next_waypoint** variable so that the agent can get information from the planner on the next direction it should go.

I chose not to include other parameters like **deadline**, or **location** simply because they will make the state space too large, thereby forcing the agent to take too long to explore all of the state space.

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

Immediately after implementing Q-learning, the agent stops behaving randomly, and it starts to find its way to the destinations.

Finally, after 1 or 2 trials, the agent is able to quickly find it's way to destination, while obeying traffic rules.

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 run it with the *alpha*, *gamma*, and the *q_init* variables. The *q_init* variable is used to initialise the Q matrix.

I obtained the values displayed in the following table:

	Rewards	Time_to_destination	Reached_destination
Alpha:0.8 Gamma:0.4 q_init:2.5	298.0	136	10
Alpha:1.0 Gamma:0.4 q_init:2.0	315.5	147	10
Alpha:1.0 Gamma:0.4 q_init:3.0	325.0	167	10
Alpha:1.0 Gamma:0.2 q_init:3.5	276.0	124	10
Alpha:0.8 Gamma:0.4 q_init:3.0	315.5	151	10
Alpha:0.8 Gamma:0.2 q_init:3.5	264.0	107	10
Alpha:0.8 Gamma:0.2 q_init:4.0	338.5	160	10
Alpha:0.6 Gamma:0.6 q_init:4.0	331.0	172	10
Alpha:0.6 Gamma:0.4 q_init:2.0	286.5	132	10

These are all parameter combinations which reached their destination 10 times. I then chose the final parameter for my agent using a greedy approach: the agent which maximised it's Rewards. This is the agent with parameters **Alpha = 0.8, Gamma = 0.2 and Init_1 = 4.0**. I chose to ignore the time_to_destination, due to my greedy nature :)