Implement a basic driving agent

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

Agent's Behavior

The agent's actions are simply random and the deadline is not enforced, thus the agent can take a short time or a long time to arriving to the target location. It does eventually make it to the target location, but it breaks traffic rules, and collides with other cars.

Identify and update state

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

I chose next_waypoint, light, oncoming, right, and left as the set of states.

- **Next_waypoint (waypoint to destination)**
  - Knowing its next step is important in learning where it is going and understanding its position relative to its final destination.
- **Light (state of traffic light: 'red' or 'green')**
  - With this, the agent can learn the basic traffic rules of the road and navigate without any collisions or illegal activity.
- **Oncoming, Right, Left (direction of car in the oncoming, right, and left lane is headed)**
  - With these states, the agent can see where a car is coming and going and avoid collisions. It can better understand it's surroundings in the environment.

As the number of states grows, the longer it will take to train and learn Q values for the states. However, it is important to have the states listed above to insure a legal collision-free cab ride.

Implement Q-Learning

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

The agent does reach its destination more often than the random method. Most of the times, the agent is able to move quickly within the deadline time towards the destination. Compared to the random method, the agent is following the next waypoint, avoiding driving on a red light or colliding with another car.

With the help of Q-learning, the agent is able to evaluate the state-action pairs, calculate the utility of sequences of action, and follow an optimal policy that will guide it to taking the best action in a certain state.

Enhance the driving agent

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 experimented with the learning rate and the discount factor of Q-Learning to achieve the final version of the agent. The number of penalties shown below are the counts of negative rewards throughtout the 100 trials.

Learning Rate (alpha)	Discount Factor (gamma)	Number of Penalties (Trials with penalties)	Passed Trials/Total Trials
0.5	0.5	16(11 trials)	96/100
0.5	0.3	17(15 trials)	1/100
0.4	0.25	14(16 trials)	99/100
0.5	0.25	10(9 trials)	100/100
0.6	0.25	13(12 trials)	27/100

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?

Yes I believe it finds an optimal policy at a learning rate of 0.5 and discount factor of 0.25. The agent reaches the destination 100% of the time. However, it incurs total of 10 penalties in nine trials so the actual number of valid passed trials is 91/100. As the agent learns, it faces some penalties in the beginning but as it goes through more trials, the agent is able to reach its final destination within the deadline and not incur any penalties.