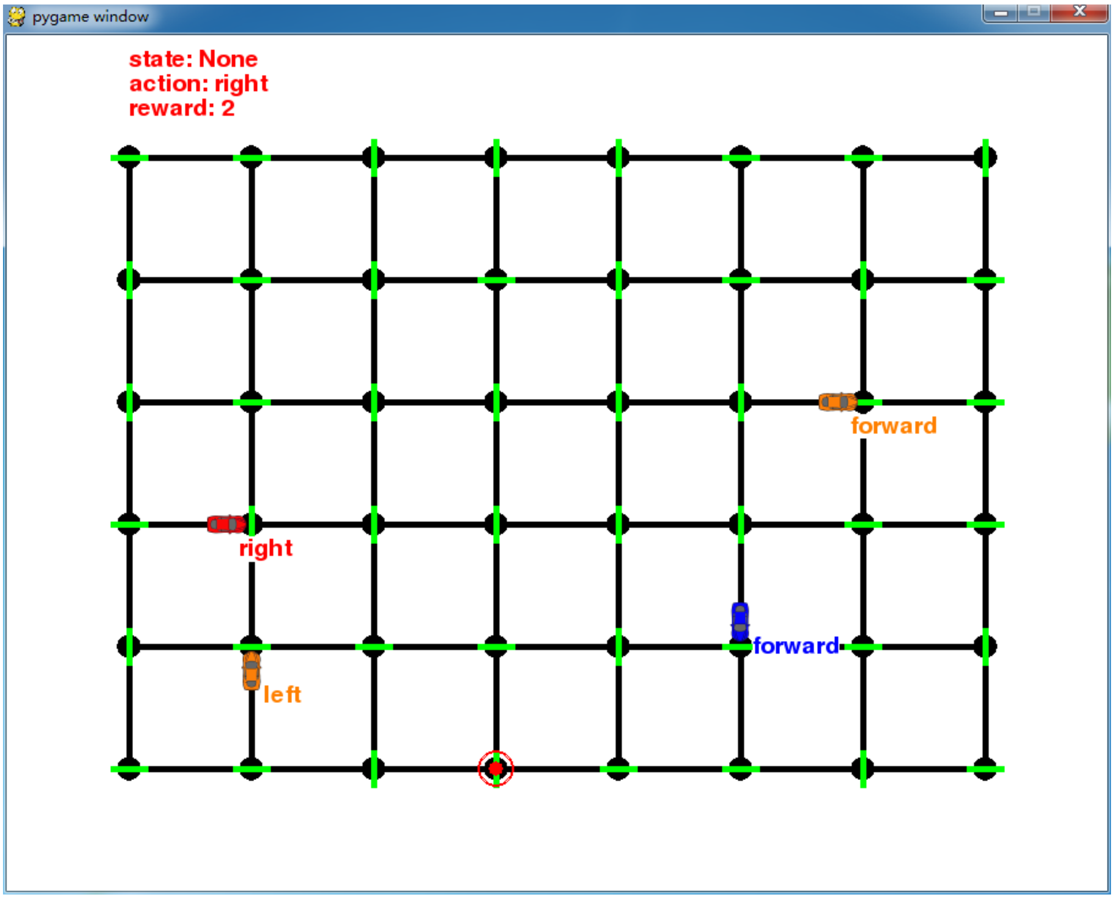
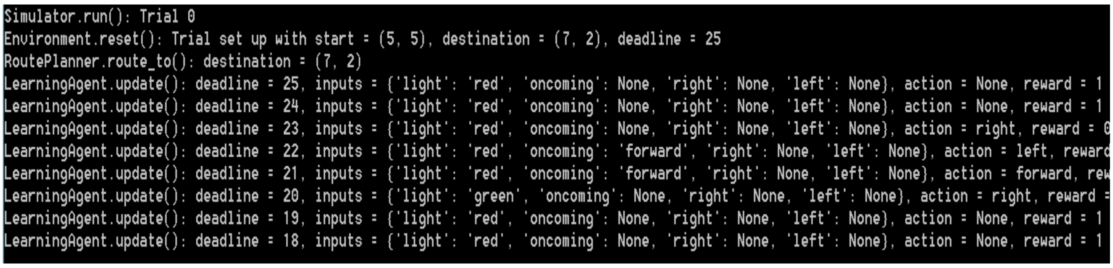
1. The running screen shot is as follow：



state: None
action: right
reward: 2



```
Simulator.run(): Trial 0
Environment.reset(): Trial set up with start = (5, 5), destination = (7, 2), deadline = 25
RoutePlanner.route_to(): destination = (7, 2)
LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1
LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1
LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0
LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = left, reward
LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, rew
LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward =
LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1
LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1
```

The red car is taking action randomly, but after a long time, it can reach the destination.

2. State includes "light", "oncoming", "right", "left" and "next_waypoint". "Light", "oncoming", "right", "left" is important, these give agent whether it will get punishment by taking some actions, ignore any of these, the car may break the traffic rules and

get negative reward. "Next_waypoint" gives the agent the right direction to go to the destination as soon as possible, without it, the car will just hang around randomly. "deadline" is not included, because as time goes on, the agent strategy won't change. But if it's included, every time the agent takes a step, it will get into a totally new state, the old state strategy will never be used, thus the agent will not learn anything useful.

3. The agent will tend to follow the traffic rules and take positive reward actions and walk to the destination. The reason the agent will follow the traffic rules is that Q learning is always choosing the best reward action, and any illegal action will get negative reward while legal actions will get non negative reward, thus illegal actions will not be chosen. "next_waypoint" is the best direction to follow to get to the destination as soon as possible. And any time the agent follow this direction without breaking any traffic rules, it will get a bonus reward, which makes this a best reward action, thus the agent will tend to choose this action and walk to the destination.

4. I have tried different combinations of alpha and gema:

| (alpha, gema) | (1, 1) | (0.1, 0.1) | (0.5, 0) |
|---|---|---|---|
| Average left time ratio | 0.616 | 0.512 | 0.561 |
| Negative reward step ratio | 0 | 0.034 | 0.017 |
| Reaching destination ratio | 0.6 | 0.8 | 0.98 |

**Average left time ratio** is calculated by this: (the deadline when reaching the destination / the deadline from the beginning) / number of reaching. This represent the speed of the agent to get to the destination.

**Negative reward step ratio**: (number of negative reward steps / number of steps to reach the destination) / number of reaching. This represent the agent's ability to obey the traffic rules.

**Reaching destination ratio**: number of reaching destination / number of trials.

We can find that (1, 1) can give good result, but it doesn't guarantee perfect destination reached, sometimes the car is always turning right and getting stuck. (0.5, 0) is a great choice, the car can find a good strategy to get to the destination very fast and achieve good **Reaching destination ratio**.

The agent's ideal policy would be, go to the destination along the shortest path and never break any traffic rules. The final driving agent almost reaches the destination and obey the traffic rules all the time , but it doesn't guarantee to follow the ideal routes every time. When there are conflict between following the ideal routes and obeying the traffic rules, the car will always choose later.