Implement a Basic Driving Agent

QUESTION: Observe what you see with the agent's behavior as it takes random actions. Does the smartcab eventually make it to the destination? Are there any other interesting observations to note?

ANSWER: The agent does eventually make it to the destination, after a very long time. In my first simulation it took 89 turns AFTER the deadline to reach the destination.

I'm not sure these are actually interesting observations, but the agent is an awful driver when random. It regularly violates the rules of the road, and has no idea where it is going at any given time. These are to be expected with completely random actions.

Inform the Driving Agent

QUESTION: What states have you identified that are appropriate for modeling the smartcab and environment? Why do you believe each of these states to be appropriate for this problem?

ANSWER: I have identified 'light', 'oncoming', and 'left' as appropriate for modeling the environment. If 'light' is 'green', we can go 'forward' or 'right', regardless of the other inputs. If 'light' is 'green' and 'oncoming' is 'None', we can go 'left'. If 'light' is 'red' and 'left' is 'None', we can go 'right'. In all other states, we should idle.

There is never a time where the 'right' input matters for our action.

OPTIONAL: How many states in total exist for the smartcab in this environment? Does this number seem reasonable given that the goal of Q-Learning is to learn and make informed decisions about each state? Why or why not?

ANSWER: There are 8 (aka 2^3) possible states that exist for the smartcab in this environment. This seems like a reasonable number, given the Q-Learning Goal. That is, we can probably use a simple table to store this data.

Implement a Q-Learning Driving Agent

QUESTION: What changes do you notice in the agent's behavior when compared to the basic driving agent when random actions were always taken? Why is this behavior occurring?

Improve the Q-Learning Driving Agent

QUESTION: Report the different values for the parameters tuned in your basic

implementation of Q-Learning. For which set of parameters does the agent perform best? How well does the final driving agent perform?

QUESTION: 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? How would you describe an optimal policy for this problem?