Smartcab project

Mention what you see in the agent's behavior. Does it eventually make it to the target location?

In this first part I see the agent is moving randomly, not following any policy. The actions are executed with no concern about the obtained reward. The agent is learning nothing. For this reason, the agent gets a lot of penalties in every trial, it's not doing any better.

Eventually, the agent reaches the destination, in most of the trials. This is because of the lack policy and time limit (there is a hard time limit from the environment to avoid the simulation run indefinitely).

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

In order to select the states, first I have to analyze the data available from the environment and the rules to drive.

There are the next variables in the environment an their possible values:

- Next waypoint: Left, Right, Forward
- Traffic light: Green, Red
- Oncoming (Front) car: Left, Right, Forward, None
- Left car: Left, Right, Forward, None
- Right car: Left, Right, Forward, None
- Time steps remaining: [Start value – hard time limit | 0]

First, we can get rid of **Time steps remaining** because the range of values for this variable can be high, therefore increasing too much the number of states.

Next, based on the rules for driving, there is no rule applied to turn right, therefore we can put **Right car** aside.

This gives me four variables to analyze and found the possible states. The easiest way to do so is create a table of all possible combinations.

States	Next way point	Traffic light	Front car	Left car
s0	Left	Green	Left	Forward
s1	Left	Green	Left	Right | Left | None
s2	Left	Green	Forward	Forward
s3	Left	Green	Forward	Right | Left | None
s4	Left	Green	Right | None	Forward
s5	Left	Green	Right | None	Right | Left | None
s6	Left	Red	Left	Forward
s7	Left	Red	Left	Right | Left | None
s8	Left	Red	Forward	Forward
s9	Left	Red	Forward	Right | Left | None
s10	Left	Red	Right | None	Forward
s11	Left	Red	Right | None	Right | Left | None
s12	Right	Green	Left	Forward
s13	Right	Green	Left	Right | Left | None
s14	Right	Green	Forward	Forward
s15	Right	Green	Forward	Right | Left | None
s16	Right	Green	Right | None	Forward
s17	Right	Green	Right | None	Right | Left | None
s18	Right	Red	Left	Forward
s19	Right	Red	Left	Right | Left | None
s20	Right	Red	Forward	Forward
s21	Right	Red	Forward	Right | Left | None
s22	Right	Red	Right | None	Forward
s23	Right	Red	Right | None	Right | Left | None
s24	Forward	Green	Left	Forward
s25	Forward	Green	Left	Right | Left | None
s26	Forward	Green	Forward	Forward
s27	Forward	Green	Forward	Right | Left | None
s28	Forward	Green	Right | None	Forward
s29	Forward	Green	Right | None	Right | Left | None
s30	Forward	Red	Left	Forward
s31	Forward	Red	Left	Right | Left | None
s32	Forward	Red	Forward	Forward
s33	Forward	Red	Forward	Right | Left | None
s34	Forward	Red	Right | None	Forward
s35	Forward	Red	Right | None	Right | Left | None

As shown in the table, all possible combinations give 36 states. This table is already reduced, otherwise, it would have have 96 states.

There are four possible values for **Front car** and **Left car** (Left, Right, Forward, None), but the table shows only three values (Left, Forward, Right | None) for the former one, and just two values for the latter (Forward, Right | Left | None). In both cases this comes from the driving rules. In the first situation, we only care about forward and left directions because those are the ones used for our agent to turn left or right respectively. The absence of a car (None) or a right direction can be treated as one value. The same is true for left car, we just care about its forward direction to turn right.

We can reduce this table even more by analyzing the 36 possible states and merge some of them.

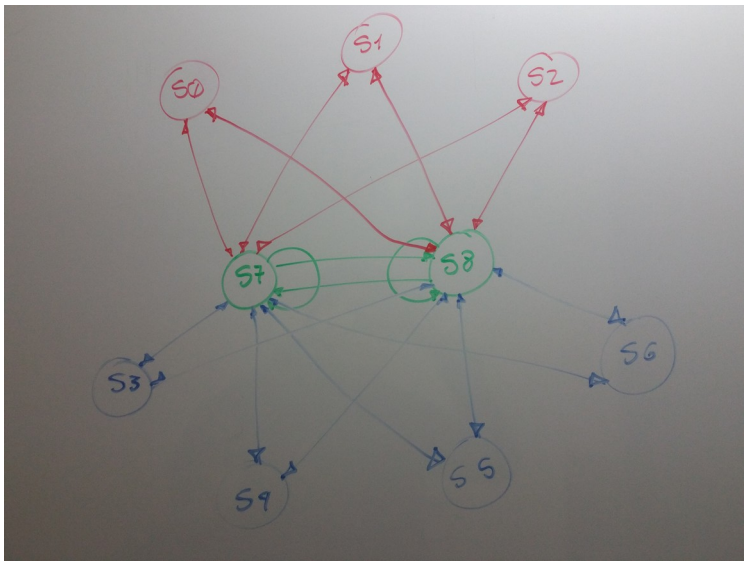
States	Next way point	Traffic light	Front car	Left car	Allows to go to next point
s0	Left	Green	Forward	Any	NO
s1	Left	Green	Right | None	Any	YES
s2	Left	Red	Any	Any	NO
s3	Right	Green	Any	Any	YES
s4	Right	Red	Any	Forward	NO
s5	Right	Red	Left	Any	NO
s6	Right	Red	Right | None	Right | Left | None	YES
s7	Forward	Green	Any	Any	YES
s8	Forward	Red	Any	Any	NO

The table is self explanatory, it basically says when the agent can or can't move to the next way point.

For instance, s0 does not allow to go to next way point because, as the driving rules states: “On a green light, you can turn left only if there is no oncoming traffic at the intersection coming straight.”

The value “Any” means any of the possible values for the cars: Right, Left, Forward, None.

Based on these states, we can have the next graph for transitions between states:



This graph is created in the following assumption: The car has to reach the destination with the minimum number of turns. The next image clarifies this assuptiom.