Project Report

You will be required to submit a project report along with your modified agent code as part of your submission. As you complete the tasks below, include thorough, detailed answers to each question *provided in italics*.

Implement a Basic Driving Agent

To begin, your only task is to get the **smartcab** to move around in the environment. At this point, you will not be concerned with any sort of optimal driving policy. Note that the driving agent is given the following information at each intersection:

* The next waypoint location relative to its current location and heading.
* The state of the traffic light at the intersection and the presence of oncoming vehicles from other directions.
* The current time left from the allotted deadline.

To complete this task, simply have your driving agent choose a random action from the set of possible actions (`None`, `'forward'`, `'left'`, `'right'`) at each intersection, disregarding the input information above. Set the simulation deadline enforcement, `enforce_deadline` to `False` and observe how it performs.

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?*

A: *The smartcab eventually makes it to the destination bit it takes a considerable amount of time. I did not note any other interesting occurrences.*

Inform the Driving Agent

Now that your driving agent is capable of moving around in the environment, your next task is to identify a set of states that are appropriate for modeling the **smartcab** and environment. The main source of state variables are the current inputs at the intersection, but not all may require representation. You may choose to explicitly define states, or use some combination of inputs as an implicit state. At each time step, process the inputs and update the agent's current state using the `self.state` variable. Continue with the simulation deadline enforcement `enforce_deadline` being set to `False`, and observe how your driving agent now reports the change in state as the simulation progresses.

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?*

A: *The state variables with the possible values I have identified for modeling the smartcab and environment are: the light which can be green or red, the presence of a car in the oncoming, left, or right lanes; their intended action which can be forward, left, right, or none; and the next waypoint which can be forward, left, or right. The deadline variable has been ignored because it would cause the state space to grow much too large for effective learning. The smartcab needs to know the state of the light so it only travels when it is allowed by traffic regulations. Similarly, it needs to know whether there are other cars around and their intended travel in order to yield properly. Finally, it must know its next waypoint in order to learn to reach its goal.*

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?*

A: *Independently, the light has 2 states, and each direction of traffic(3 total) has 4 possible states, and the next waypoint has 3 states, which multiplied together give 384 states given the earlier identified states. This number of states seems like it would make Q learning difficult to learn and make informed decisions regarding each possible state given the deadline imposed in this situation means it may have limited information on each state.*

Implement a Q-Learning Driving Agent

With your driving agent being capable of interpreting the input information and having a mapping of environmental states, your next task is to implement the Q-Learning algorithm for your driving agent to choose the *best* action at each time step, based on the Q-values for the current state and action. Each action taken by the **smartcab** will produce a reward which depends on the state of the environment. The Q-Learning driving agent will need to consider these rewards when updating the Q-values. Once implemented, set the simulation deadline enforcement `enforce_deadline` to `True`. Run the simulation and observe how the **smartcab** moves about the environment in each trial.

The formulas for updating Q-values can be found in this video.

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?*

A: The agent goal oriented behavior. After a few trials, it has learned fairly well to follow the planned path and avoid moves that result in adverse rewards. This behavior arises from the Q learning algorithm allowing the model to adapt to the rewards it has received in the past to improve the decision it makes in the future.

Improve the Q-Learning Driving Agent

Your final task for this project is to enhance your driving agent so that, after sufficient training, the **smartcab** is able to reach the destination within the allotted time safely and efficiently. Parameters in the Q-Learning algorithm, such as the learning rate (`alpha`), the discount factor (`gamma`) and the exploration rate (`epsilon`) all contribute to the driving agent’s ability to learn the best action for each state. To improve on the success of your **smartcab**:

* Set the number of trials, `n_trials`, in the simulation to 100.
* Run the simulation with the deadline enforcement `enforce_deadline` set to `True` (you will need to reduce the update delay `update_delay` and set the `display` to `False`).
* Observe the driving agent’s learning and **smartcab’s** success rate, particularly during the later trials.
* Adjust one or several of the above parameters and iterate this process.

This task is complete once you have arrived at what you determine is the best combination of parameters required for your driving agent to learn successfully.

The table below tracks the value of mistakes made, the steps left before the deadline when the agent reached its destination, the deadline of steps to take, and the percent of the deadline used to reach the goal for the last ten trials given several values of parameters.

Trial 1	*A*	*0.7*	*G*	*0.5*	*E*	*0.8*
	Mistakes	*steps left*	*deadline*	*% Time Taken*		
90	*0*	*11*	*20*	*45%*		
91	*-0.5*	*10*	*30*	*67%*		
92	*0*	*17*	*20*	*15%*		
93	*0*	*20*	*30*	*33%*		
94	*0*	*28*	*40*	*30%*		
95	*-1*	*26*	*50*	*48%*		
96	*0*	*21*	*25*	*16%*		
97	*-1*	*25*	*50*	*50%*		
98	*0*	*11*	*20*	*45%*		
99	*0*	*24*	*35*	*31%*		
	-0.25			*38%*		
Trial 2	*A*	*0.7*	*G*	*0.4*	*E*	*0.8*
	Mistakes	*steps left*	*deadline*	*% Time Taken*		
90	*-1*	*13*	*25*	*48%*		
91	*0*	*15*	*20*	*25%*		
92	*0*	*17*	*30*	*43%*		
93	*0*	*9*	*25*	*64%*		
94	*0*	*29*	*40*	*28%*		
95	*-1*	*12*	*25*	*52%*		
96	*0*	*37*	*55*	*33%*		

97	*0*	*14*	*25*	*44%*		
98	*-2*	*25*	*40*	*38%*		
99	*0*	*31*	*45*	*31%*		
	-0.4			*41%*		
Trial 3	*A*	*0.8*	*G*	*0.5*	*E*	*0.8*
	Mistakes	*steps left*	*deadline*	*% Time Taken*		
90	*0*	*0*	*35*	*100%*		
91	*-0.5*	*0*	*20*	*100%*		
92	*0*	*0*	*60*	*100%*		
93	*0*	*0*	*30*	*100%*		
94	*0*	*0*	*20*	*100%*		
95	*0*	*0*	*40*	*100%*		
96	*0*	*0*	*20*	*100%*		
97	*0*	*0*	*25*	*100%*		
98	*0*	*0*	*30*	*100%*		
99	*0*	*0*	*30*	*100%*		
	-0.05			*100%*		
Trial 4	*A*	*0.6*	*G*	*0.5*	*E*	*0.8*
	Mistakes	*steps left*	*deadline*	*% Time Taken*		
90	*0*	*25*	*40*	*38%*		
91	*0*	*20*	*50*	*60%*		
92	*0*	*11*	*20*	*45%*		
93	*0*	*12*	*20*	*40%*		
94	*0*	*13*	*25*	*48%*		

95	*0*	*13*	*30*	*57%*		
96	*0*	*15*	*25*	*40%*		
97	*-0.5*	*16*	*45*	*64%*		
98	*0*	*21*	*30*	*30%*		
99	*-1*	*12*	*20*	*40%*		
	-0.15			*46%*		
Trial 5	*A*	*0.55*	*G*	*0.5*	*E*	*0.8*
	Mistakes	*steps left*	*deadline*	*% Time Taken*		
90	*0*	*18*	*30*	*40%*		
91	*0*	*9*	*25*	*64%*		
92	*0*	*18*	*35*	*49%*		
93	*0*	*18*	*25*	*28%*		
94	*-2*	*19*	*25*	*24%*		
95	*0*	*1*	*25*	*96%*		
96	*0*	*21*	*30*	*30%*		
97	*0*	*33*	*45*	*27%*		
98	*-0.5*	*7*	*20*	*65%*		
99	*0*	*8*	*35*	*77%*		
	-0.25			*50%*		
Trial 6	*A*	*0.6*	*G*	*0.6*	*E*	*0.8*
	Mistakes	*steps left*	*deadline*	*% Time Taken*		
90	*-1*	*16*	*30*	*47%*		
91	*0*	*12*	*30*	*60%*		
92	*0*	*25*	*40*	*38%*		

93	***-1***	***9***	***25***	***64%***		
94	***0***	***15***	***20***	***25%***		
95	***0***	***13***	***30***	***57%***		
96	***0***	***13***	***20***	***35%***		
97	***0***	***5***	***25***	***80%***		
98	***0***	***17***	***30***	***43%***		
99	***-1***	***28***	***40***	***30%***		
	-0.3			***48%***		

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?*

A:

Alpha: 0.6

Gamma: 0.5

Epsilon: 0.8

During the last ten trials the agent averages -0.15 worth of mistakes while reaching its destination each time in an average of 46% of the time allotted.

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?*

A: The agent is close to finding the optimal policy. It still incurs some penalties but uses less than half the allotted time to reach the destination. More specific tuning could probably get it even closer. Given the problem set, an optimal policy would ensure 100% safety of the occupants and others on the road while reaching the destination in a reasonable amount of time. This policy for safety may result in the smartcab taking a longer time to reach its destination however it should not *cause* any accidents or commit any driving errors. This cautious approach can be seen in many “smartcab” developments today erring on the side of caution.

Compared to this optimal policy, my agent does make errors, likely due to the limited amount of time it is given to explore the state space compared to the millions of miles logged by current autonomous car efforts.   Beyond driving errors, the agent does not seem to exhibit much tendency to taking a longer route for more rewards, indicating it is following the planned path.