**Udacity Smartcab project**

Tomoaki Tsuzuki

**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:*** *I have run 100 times with random actions. Surprisingly the agent reached the goal more than 50% of the time with* ***enforce\_deadline*** *to be* ***False****. It is 61/100 times. The average steps to take to reached destination is about 55 steps.*

*The graph below shows how random smartcab performed.*

*Blue line shows if the agent reached to destination (0 = failed because of hard time limit, 1 = reached to goal). Orange line shows time steps to take.*

*Obviously there is no trend for both success or fail and time steps.*

*Hopefully with Q-learning implemented, it shows trend of getting better and better.*

### 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 selected below as the state variables.*

* *light {red, green}*
* *oncoming {None, forward, left, right}*
* *left {None, forward, left, right}*
* *right {None, forward, left, right}*
* *next\_waypoint {None, forward, legt, right)*

*The state is combination of those state variables. So it is total of 512 states.*

*Since possible actions to take is {None, forward, left, right}, my Q-matrix would be the size of 512 x 4.*

*There is one more state variables that can be used, which is dead line.*

*However, it greatly increases the size of Q-matrix because deadline variable can take many states (distance between start and destination times 5).*

*For the agent to learn within the reasonable amount of time, I did not include deadline as state variables.*