# Reinforcement Learning

## Project: Train a Smartcab How to Drive

## Project Overview

In this project you will apply reinforcement learning techniques for a self-driving agent in a simplified world to aid it in effectively reaching its destinations in the allotted time. You will first investigate the environment the agent operates in by constructing a very basic driving implementation. Once your agent is successful at operating within the environment, you will then identify each possible state the agent can be in when considering such things as traffic lights and oncoming traffic at each intersection. With states identified, you will then implement a Q-Learning algorithm for the self-driving agent to guide the agent towards its destination within the allotted time. Finally, you will improve upon the Q-Learning algorithm to find the best configuration of learning and exploration factors to ensure the self-driving agent is reaching its destinations with consistently positive results.

## Description

In the not-so-distant future, taxicab companies across the United States no longer employ human drivers to operate their fleet of vehicles. Instead, the taxicabs are operated by self-driving agents, known as \*smartcabs\*, to transport people from one location to another within the cities those companies operate. In major metropolitan areas, such as Chicago, New York City, and San Francisco, an increasing number of people have come to depend on \*smartcabs\* to get to where they need to go as safely and reliably as possible. Although \*smartcabs\* have become the transport of choice, concerns have arose that a self-driving agent might not be as safe or reliable as human drivers, particularly when considering city traffic lights and other vehicles. To alleviate these concerns, your task as an employee for a national taxicab company is to use reinforcement learning techniques to construct a demonstration of a \*smartcab\* operating in real-time to prove that both safety and reliability can be achieved.

## Software Requirements

This project uses the following software and Python libraries:

- [Python 2.7](https://www.python.org/download/releases/2.7/)

- [NumPy](http://www.numpy.org/)

- [pandas](http://pandas.pydata.org/)

- [matplotlib](http://matplotlib.org/)

- [PyGame](http://pygame.org/)

If you do not have Python installed yet, it is highly recommended that you install the [Anaconda](http://continuum.io/downloads) distribution of Python, which already has the above packages and more included. Make sure that you select the Python 2.7 installer and not the Python 3.x installer. `pygame` can then be installed using one of the following commands:

Mac: `conda install -c https://conda.anaconda.org/quasiben pygame`

Windows: `conda install -c https://conda.anaconda.org/tlatorre pygame`

Linux: `conda install -c https://conda.anaconda.org/prkrekel pygame`

This project contains five files and one folder:

* `agent.py`: This is the main Python file where you create a learning agent and driving function for running the simulation
* `environment.py`: This Python file creates the \*smartcab\* environment.
* `planner.py`: This Python file creates a high-level planner for the agent to follow towards a set goal.
* `simulation.py`: This Python file creates the simulation and graphical user interface.
* `visual.py`: This Python file provides supplementary visualization for the analysis.
* `/images/`: This folder contains various images of cars to be used in the graphical user interface.

### Running the Code

In a terminal or command window, run one of the following commands:

`python agent.py`

This will run the `agent.py` file and execute your implemented agent code into the environment.

## Definitions

### Environment

The \*smartcab\* operates in an ideal, grid-like city (similar to New York City), with roads going in the North-South and East-West directions. Other vehicles will certainly be present on the road, but there will be no pedestrians to be concerned with. At each intersection there is a traffic light that either allows traffic in the North-South direction or the East-West direction. U.S. Right-of-Way rules apply:

- On a green light, a left turn is permitted if there is no oncoming traffic making a right turn or coming straight through the intersection.

- On a red light, a right turn is permitted if no oncoming traffic is approaching from your left through the intersection.

To understand how to correctly yield to oncoming traffic when turning left, you may refer to [this official drivers? education video](https://www.youtube.com/watch?v=TW0Eq2Q-9Ac), or [this passionate exposition](https://www.youtube.com/watch?v=0EdkxI6NeuA).

### Inputs and Outputs

Assume that the \*smartcab\* is assigned a route plan based on the passengers? starting location and destination. The route is split at each intersection into waypoints, and you may assume that the \*smartcab\*, at any instant, is at some intersection in the world. Therefore, the next waypoint to the destination, assuming the destination has not already been reached, is one intersection away in one direction (North, South, East, or West). The \*smartcab\* has only an egocentric view of the intersection it is at: It can determine the state of the traffic light for its direction of movement, and whether there is a vehicle at the intersection for each of the oncoming directions. For each action, the \*smartcab\* may either idle at the intersection, or drive to the next intersection to the left, right, or ahead of it. Finally, each trip has a time to reach the destination which decreases for each action taken (the passengers want to get there quickly). If the allotted time becomes zero before reaching the destination, the trip has failed.

### Rewards and Goal

The \*smartcab\* will receive positive or negative rewards based on the action it as taken. Expectedly, the \*smartcab\* will receive a small positive reward when making a good action, and a varying amount of negative reward dependent on the severity of the traffic violation it would have committed. Based on the rewards and penalties the \*smartcab\* receives, the self-driving agent implementation should learn an optimal policy for driving on the city roads while obeying traffic rules, avoiding accidents, and reaching passengers? destinations in the allotted time.