Task 1: Implement a basic driving agent

Implement the basic driving agent, which processes the following inputs at each time step:

* Next waypoint location, relative to its current location and heading,
* Intersection state (traffic light and presence of cars), and,
* Current deadline value (time steps remaining),

And produces some random move/action (None, 'forward', 'left', 'right'). In your report, mention what you see in the agent’s behaviour. Does it eventually make it to the target location?

The implementation was made on the “*update*” method of the “**LearningAgent**” class on the apent.py file. That method uses a function called “get\_next\_waypoint\_given\_location” from the Agent class, that, given the agent’s locations, it randomly select the next action. The only caveat is that I wrote this function to prevent the agents from crossing from one side to the other of the map, so if the agent is facing north at position (1, 3), its only available options are left and right. This function is also shared with the dummy agents.

Also, to avoiding code duplication, I moved the code that checks if an action is ok from the “*update*” method on the “**DummyAgent**” class to a function called “*check\_if\_action\_is\_ok*” inside the “**Agent**” class, so I can use it on the “*update*” method on the “**LearningAgent**” class.

As expected, the agent moves around the map randomly. Eventually it may reach the destination but it is only a matter of chance.

Task 2: Identify and Update States