Here are the types of the inputs and outputs of the smart cap from the Udacity Project Description.

**Inputs:**

1. The next waypoint
2. Traffic light for its direction of movement
3. Other cars’ location and their next movement directions when they are near the agent
4. deadline

**Outputs:**

1. Agent’s action – stay put, move one block forward, one block left or one block right

# Implement a basic driving agent

<agent.py>

# Gather inputs

self.next\_waypoint = self.planner.next\_waypoint() # from route planner, also displayed by simulator

inputs = self.env.sense(self)

deadline = self.env.get\_deadline(self)

# TODO: Update state

self.state = (inputs, self.next\_waypoint, deadline)

# TODO: Select action according to your policy

action = random.choice([None, 'forward','left','right'])

A basic driving agent with random action regardless of its surrounding conditions was implemented. The reward system was verified based on the next waypoints, traffic lights, neighboring cars and the agent’s (random) action. As shown the Figure 1, the agent received the reward of -1 when it moved forward when the light was red.

The agent very rarely does arrive at the destination, but since the movements are completely random, it should not be called ‘smart’ cab. It did not take into account any of the Inputs; the traffic light, the other cars, the next waypoint information or the time steps (deadline).

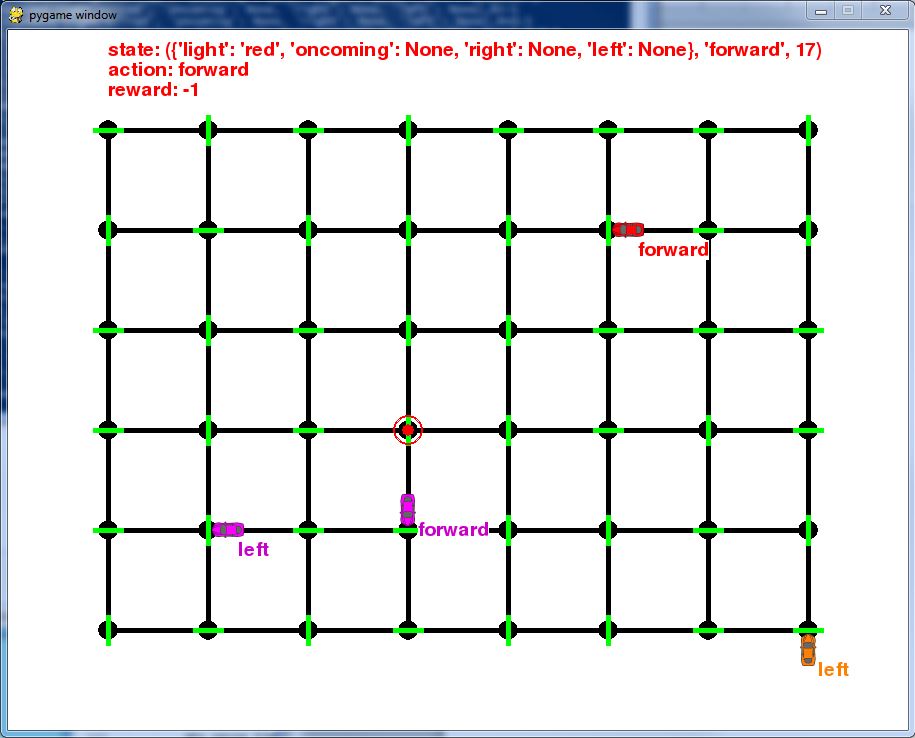


Figure Basic driving agent

# Identify and update state

The state of the agent was initiated and updated with the pre-specified variables in the script (self.next\_waypoint, inputs, deadline).