```
import random

class RoutePlanner(object):
    """Silly route planner that is meant for a perpendicular grid network."""

    def __init__(self, env, agent):
        self.env = env
        self.agent = agent
        self.destination = None

    def route_to(self, destination=None):
        self.destination = destination if destination is not None else
random.choice(self.env.intersections.keys())
        print "RoutePlanner.route_to(): destination = {}".format(destination)  # [debug]

    def next_waypoint(self):
        location = self.env.agent_states[self.agent]['location']
        heading = self.env.agent_states[self.agent]['heading']
        delta = (self.destination[0] - location[0], self.destination[1] - location[1])
        if delta[0] == 0 and delta[1] == 0:
            return None
        elif delta[0] != 0:  # EW difference
            if delta[0] * heading[0] > 0:  # facing correct EW direction
                return 'forward'
            elif delta[0] * heading[0] < 0:  # facing opposite EW direction
                return 'right'  # long U-turn
            elif delta[0] * heading[1] > 0:
                return 'left'
            else:
                return 'right'
        elif delta[1] != 0:  # NS difference (turn logic is slightly different)
            if delta[1] * heading[1] > 0:  # facing correct NS direction
                return 'forward'
            elif delta[1] * heading[1] < 0:  # facing opposite NS direction
                return 'right'  # long U-turn
            elif delta[1] * heading[0] > 0:
                return 'right'
            else:
                return 'left'
```