

## Project 1

### The Game:

The game was inspired by the difficulties in getting around in a large metropolis like NY City. In its simplified form, it consists in going from one place to another in a town by using the available modes of transportation in the shortest possible time. More precisely, a player is supposed to start at one place in a town and get to another predefined place by either walking, or taking a cab. Cabs can be stuck in traffic jams. Other features can be added if time permits.

### Rules:

We have a town with a rectangular grid of streets and avenues.

Avenues go north/south, streets are east/west.

All streets and avenues are one way.

There are lights at every intersection.

Cabs (and if time permits busses) travel the town.

Traffic jams pop at random times in random places

Traffic jams can be severe or benign and the severity determines by how much taxis slow down.

Walkers are not affected by jams, but of course they are slower. They do not jay walk and do not cross the street if the light is green for cars.

### Goal:

The goal is to go from lower left position to higher right position in the fastest way possible.

The user will need to decide how to get from origin to target by switching between walking and taking a taxi based on the lights, the traffic jams, and the availability of taxis.

### Assumptions:

All numbers are integers. Changes are driven by time\_step. Positions are defined by a list (or tuple) of 2 integers. Lights switch at every time step.

A block is the distance between 2 consecutive streets or avenues. Speeds are numbers of block per time step. Traffic jams can extend for one or more blocks.

### Classes:

class map:

**Main class. Describes the entire town with all streets/avenues. At each time steps updates lights, positions, and state of each class**

Methods:

```
Update(play_game.next_step)
draw() draw map of town with all elements in it returns()
```

Attributes:

```
road[]
lights[]
traffic_jams[]
player
```

taxis[]

**class transport:**

**Methods:**

**Change\_direction(self,direction)**  
    **Stop()**  
    **Time\_step()**  
    **switch\_transport(self, player)**  
        **player.set\_transportation(self)**

**attributes:**

**x\_position**  
    **y\_position**  
    **direction**  
    **speed**

**class taxi(transport):**

**describe position and other attributes of every taxi**

**methods:**

**discharge()**  
        **self.availability = "free"**  
    **return()**

**Attribute:**

**availability**  
    **destination**  
    **in\_jam**

**class walk(transport):**

**describe walking attributes**

**no methods**

**attributes:**

**class light:**

**Traffic light. I will assume that all traffic lights have the same frequency, switch from red to green (no yellow). "color" represents the color in the north/south direction.**

**methods:**

**change\_color(self)**  
        **return(color)**

**attributes:**

**x\_position**  
    **y\_position**  
    **color**  
    **frequency = 1**

**class traffic\_jam**

**Pop at random time steps and random positions. Can be severe/benign/non-existent.**

**Benign would reduce speed of taxis by 20%, severe by 50%**

method:

```
    form(position)
        return(severity)
    disperse()
        return()
```

Attributes:

```
x_position
y_position
severity
```

**class player**

**This is the player**

methods:

```
    get_transportation()
        return(taxi or walk)
    set_transportation(taxi or walk)
        return()
    has_arrived()
        return(True or False)
```

attributes

```
x_position
y_position
speed
direction
transportation
```

**class play\_game:**

Methods:

```
    Next_action(player.x_position, player.y_position)
        get next action from user
        return next_action or end game.
```

attributes:

```
    time_elapsed
```