Prompt: Traffic Signal Simulator

Problem Description

You are tasked with creating a traffic flow simulation program. Your client for this project is Professor E, who teaches civil engineering at Oregon State University. One of the courses she teaches has a section on traffic signal timing, and according to her, this is a particularly challenging subject for her students. In short, traffic signal timing involves determining the amount of time that each of an intersection's traffic lights spend being green, yellow, and red, in order to allow cars to flow through the intersection from each direction in a fluid manner. In the ideal case, the amount of time that people spend waiting is minimized by the chosen settings for a given intersection's traffic lights. This can be a very subtle matter: changing the timing at a single intersection by a couple of seconds can have far-reaching effects on the traffic in the surrounding areas.

There is a great deal of theory on this subject, but Professor E. has found that her students find the topic quite abstract. She wants to provide them with some software that they can use to "play" with different traffic signal timing schemes, in different scenarios. She anticipates that this will allow her students to learn from practice, by observing first-hand some of the patterns that govern the subject.

Requirements

The following broad requirements should be followed when implementing this system:

1. Students using the program must be able to create a visual map of an area, laying out roads in a pattern of their choosing. The resulting map need not be complex, but should allow for roads of varying length to be placed, and different arrangements of intersections to be created. Your approach should accommodate at least six separate intersections on the map, if not more.

2. Students must be able to describe the behavior of the traffic lights at each of the intersections. It is up to you to determine what the exact interaction will be, but a variety of sequences and timing schemes should be allowed. Your approach should also be able to accommodate left-hand turns protected by left-hand green arrow lights. In addition:

    a. Combinations of individual signals that would result in crashes should not be allowed.

b. Every intersection on the map must have traffic lights (there are no stop signs, overpasses, or other variations). All intersections will be 4-way: there are no "T" intersections, nor one-way roads.

c. Students must be able to design each intersection with or without the option to have sensors that detect whether any cars are present in a given lane. The intersection's lights' behavior should be able to change based on the input from these sensors, though the exact behavior of this feature is up to you.

3. Based on the map created, and the intersection timing schemes, the students must be able to simulate traffic flows on the map. The traffic levels should be conveyed visually to the user in a real-time manner, as they emerge in the simulation. The current state of the intersections' traffic lights should also be depicted visually, and updated when they change. For example, you may choose to depict individual cars, or to use a more abstract representation.

4. Students should be able to change the traffic density that enters the map on a given road. For example, it should be possible to create a busy road, or a seldom used one, and any variation in between. How exactly this is declared by the user and depicted by the system is up to you.

Broadly, the tool should encourage students to explore multiple alternative approaches. Students should be able to observe any problems with their map's timing scheme, alter it, and see the results of their changes on the traffic patterns.

This program is not meant to be an exact, scientific simulation, but aims to simply illustrate the basic effect that traffic signal timing has on traffic. If you wish, you may assume that you will be able to reuse an existing software package that provides relevant mathematical functionality such as statistical distributions, random number generators, and queuing theory.

You may add additional features and details to the simulation, if you think that they would support these goals.

Desired Outcome

Your task is to implement the **non-graphical portion** of the traffic simulator program; you do not have to implement all of the program. Your implementation work should focus on two main tasks:

1. You must implement the basic data structures and associated APIs through which the program will represent the state of the simulation (e.g., data structures that capture cars, roads, intersections, lights, etc.). You should think very carefully about the scenarios

presented in the above, and whether your data structures can capture all aspects of the simulation.

2. You should implement a basic algorithm that advances the state of the simulation one ‘clock tick’, that is, advances all elements of the simulation one step (e.g., move a car forward, change a light if it is necessary).