Mark Klick & Simon Puzankov

Downtown Traffic Simulator

**Requirements specification:**

The downtown traffic simulator will consist of several different data structures that will model the inflow of traffic visiting downtown in CS273ville. The program will consist of several classes. The simulator itself will be a class containing the clock, all the roads in the town, and the school and bank. This class will be responsible for updating each of the roads etc. However, to further explain the design of the simulation, the data structures representing the visitors and roads will be explained.

A visitor will be modeled using a struct data type and will consist of member variables that will record times in which the visitor was driving on a specific road. Also each visitor will have a road map represented by a stack of integers; this will allow the visitor to navigate to the bank or school and back.

The class representing the inflow of traffic will be a queue of cars and will have member variables like the arrival rate; which will be specific to each user of the simulation. Also this entry queue will read in names from a text file and initialize the visitor (car) objects.

The abstract base class that will be used to model a road will have member variables consisting of 2 feeding queues and 2 departure queues which will represent the flow of traffic in and out of either side of each road that will be implemented. The actual roads will be modeled using a multi-map with the visitor’s time left on that specific road as the key and the visitors themselves as the data value. Also the map will have capacity to model the road capacity value extracted from the user input. The time efficiency of searching through a map is O(log n) , and this will allow for the multiple visitors with zero time left to the road to be removed and placed into the departure queue. Finally, within each road class there will a method implemented to essentially connect all the roads together by setting each of the roads feed and departure queues according to the map of downtown CS273ville.

Two unique derived classes for the Jackson and Travis roads must also be implemented and will extend the road base class constructor and the hookup roads method. Due to the fact that the visitors enter from both of these roads, two separate arrival queues will have to implemented as a member variable for each of these input road classes to successfully model visitors arriving to the city. These roads will also have an extra feed queue and departure queue to account for people arriving and leaving the city via two roads i.e. the two intersections in the downtown. The third departure queue representing the exit from downtown will feed into a directory queue which will keep track of all the visitors of downtown CS273ville.

After the program has received user input as to the total time for the simulator to run and what the arrival rate of visitors and capacity of the roads are, the program will initialize a simulator object and subsequently initialize all the roads, bank, and school objects will also be initialized taking into account the parameters of the simulation. With each update of the clock the visitors will navigate from the two entry roads, Jackson and Travis, to their specific destination they wish to visit using their road map. If the roads are full, the visitors patiently wait the end of the road they were just traveling on until the road has room for more cars. The average time will be calculated for each visitor’s journey through CS273ville. The program will then display a menu enabling the user to look up each of the visitors and search for a visitor by name.

**Use Cases:**

**Simulator Class:**

**Enter\_Data**

|  |  |  |
| --- | --- | --- |
| **Step** | **User Input** | **System Response** |
| **1** | **User enters arrival rate** |  |
| **2** | **User enters road capacity** |  |
| **3** | **User enters the total time of the simulation** |  |
| **4** |  | **These values are then used in the simulator’s constructor to initialize the model of downtown** |

**Update**

|  |  |  |
| --- | --- | --- |
| **Step** | **User Input** | **System Response** |
| **1** | **The value for total time** |  |
| **2** |  | **Increments the clock and calls update function for each road which models the movement of each visitor throughout downtown** |

**searchResults**

|  |  |  |
| --- | --- | --- |
| **Step** | **User Input** | **System Response** |
| **1** | **Inputs a name** |  |
| **2** |  | **System returns the statistics for the person from the CityRecord map or nothing if name not found** |

**UML:** 