

Semestral assignment

Traffic simulation

Assignment

Code program, which simulates traffic in the city. Input to the program is map of roads in proper format, list of vehicles with their route plans. Each vehicle will be represented with own thread and will has its own speed, acceleration, minimal distance from next vehicle, etc. Program will try to find optimal setting for lights. There should be proper user interface for setting global parameters as maximal speed in the city, speed of switching lights, etc. And also graphical representation of traffic and graphs for average waiting time, average waiting for green light, average speed in the city.

Solution

Input

Input to the program is provided by 5 files, which defines buildings, crossroads, turns, roads (streets) and vehicles. Loading of these files is done on the start of program with assistance of dialog window. Files must be uploaded in given order, from up to bottom.

Buildings

First line is reserved for dimensions of the city. First is width, then height. Next lines are made of 4 numbers: X-coordinate and Y-coordinate of upper left corner of the building, its width and height.

Crossroads

Crossroads are also defined by 4 numbers: X-coordinate and Y-coordinate of upper left corner of the crossroad, type of crossroad, initial time for switching light in milliseconds.

Each crossroad is made of light, which let go all vehicles from one way to every possible road, they want (if there is space for them).

Type of crossroad:

3 way crossroads are from 0 to 3, full crossroad (4 way) is 4.

- 0 - Missing northern road
- 1 - Missing western road
- 2 - Missing southern road
- 3 - Missing eastern road

Turns

Turns are implemented very similar to crossroads, with the difference, that turns let go vehicles through all the time (without waiting for green). Each turn is defined by 2 numbers: X-coordinate and Y-coordinate of upper left corner of the turn.

Roads (streets)

Each street is defined with 3 numbers: first two are IDs of turns or crossroads which are on the ends of the street, third number is, if the street is north-south (1) or west-east (2).

Street begins (from its point of view) on the southern or eastern end and ends on the northern or western end. If I want to put on some end crossroad, write number of the line in the crossroads file minus 1 (so we start from 0), where the crossroad we want to link with street is. If I want to put on some end turn, write a number of line, where the turn is in the turns file, with the negative sign.

Vehicles

Each vehicle is defined with at least 5 numbers: maximal speed, minimal distance from next vehicle, acceleration, initial street (the vehicle will always start on south or east of the street).

Fifth and every other number is defining route plan. If the fifth is -1, than vehicle will travel infinitely and randomly. Otherwise write numbers where the vehicle should turn next (if it is not possible - the street doesn't exists there, vehicle will end there)

Route plan numbers

- 0 - north
- 1 - west
- 2 - south
- 3 - east

Simulation

After loading all files correctly, you can run the simulation with button "Simulate". In the middle of newly appeared window is the city with all vehicles, buildings and lights. On the right are graphs and on the left is list of traffic lights with statistics and slider, which allows to manually change the time for switching light..

Implementation

Reading files is done in the class Reader, which manages dialog window for file input, validation of files and giving data to the City.

Simulation is handled by class City, which renders frame and all objects in it with help of its inner class CityPanel. City also handles manual input of maximal speed in the city and manual input of time for switching lights.

Each vehicle in the simulation is object of Vehicle class, its movement is handled by VehicleNode, which runs as separate thread for each vehicle.

Vehicles are moved in the queue, which is implemented as linked list, where each vehicle has pointer to vehicle in front and behind him. Linked list is implemented in Queue class. Every street has 2 queues, one for each direction.

Crossroads are object of CrossRoad class, each crossroad has its own thread and if some direction is free to go (has a green light), it transfers vehicles from this direction to directions, where the vehicles want to go. Correct switching the light is managed by object of class Light.

Program has helper classes for keeping positions, dimensions, directions and other contacts. I have also used code from StackOverflow, to render statistics graphs easily.