BLG252E Object Oriented Programming Project 2

1. Objective

The objective of this project is to enhance the traffic simulator you implemented in Project 1.

2. Requirements

The project will be implemented using SFML Graphics and Multimedia library. The library binary and header files will be provided to you.

If you use a Windows machine, you may use Dev-C++ as your build environment. For Mac and Linux, you can use a suitable C++ compiler such as gcc. No changes are required to your build environment from Project 1.

3. Project Description

The project will build upon the previous part (see previous project description for more information). In this second part, you will run the simulation with at least **6 vehicles**. The vehicle class will be modified as below.

3.1 Classes

In addition to the classes implemented in Project 1, you should implement the following classes:

1. **TrafficLight**

Private Members:

- **float x**; //x coordinate of the traffic light
- **float y**; //y coordinate of the traffic light
- **float dir**; //direction of the traffic light (determines the orientation of the traffic light on the map)
- **tLightState state**; //current state of the light (either green or red). tLightState should be an enum
- **sf::Texture redTexture**; //texture for the traffic light with red turned on
- **sf::Texture greenTexture**; //texture for the traffic light with green turned on
- **sf::Sprite sprite**; //sprite for the traffic light
- **TrafficLight *next**; //pointer to the next traffic light in the traffic light group

Public Methods:

TrafficLight(float x, float y, float dir, tLightState state)

Constructor for the TrafficLight class

parameters:

- **x**: x coordinate of the traffic light
- **y**: y coordinate of the traffic light
- **dir**: traffic light direction, i.e., orientation
- **state**: initial state of the traffic light

void getPosition(float &x, float &y, float &dir)

Returns the position and the direction of the traffic light

parameters:

- **x**: x coordinate of the traffic light
- **y**: y coordinate of the traffic light
- **dir**: traffic light direction, i.e., orientation

void draw()

Draws the traffic light to screen

tLightState getState()

Returns the current traffic light state

void setState(tLightState state)

Sets the traffic light state

parameters:

- **state**: State to set the traffic light to

2. TrafficLightGroup

Private Members:

- **TrafficLight *head**; //Pointer to the head of the traffic light in the group (hint: the group should be a linked list of traffic lights)
- **TrafficLight *greenLight**; //Pointer to the current traffic light to be turned to green
- **float time**; //Current time elapsed in seconds
- **float duration**; //Period of switching to the next traffic light to make it green

Public Methods:

TrafficLightGroup(float duration)

TrafficLightGroup constructor

parameters:

- **duration**: TrafficLight switching period

void add(TrafficLight *light)

Adds a new traffic light to the group

parameters:

- **light**: Pointer to the traffic light object to be added to the group

void simulate(float timestep)

Runs traffic light simulation by advancing the traffic light group timer by timestep

If the traffic light group timer reaches the switching period (i.e., duration member variable) of the group, the next light in the group is turned into green while the others are turned into red

parameters:

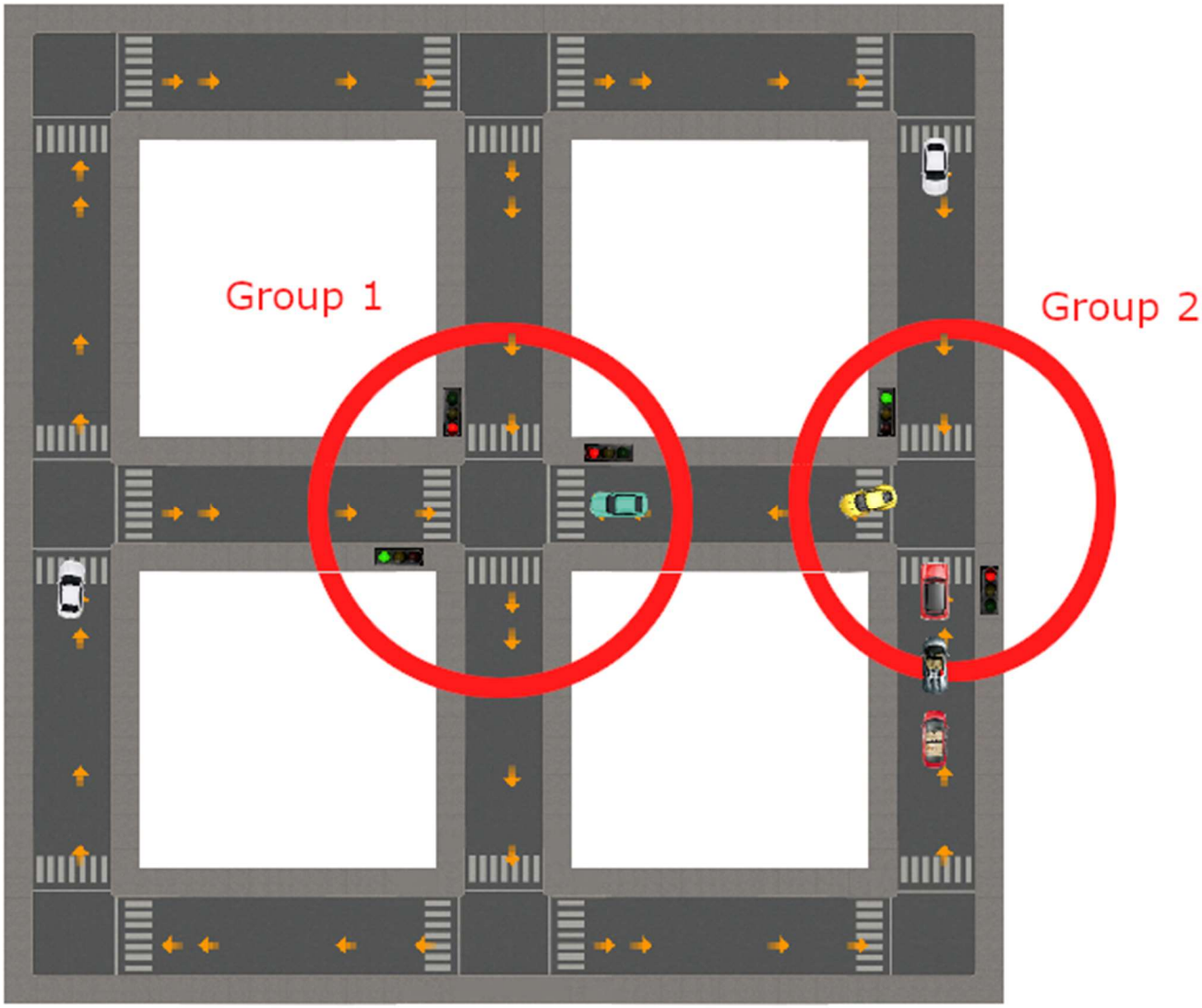
- **timestep**: timestep to be added to the current time

A traffic light group is to be instantiated at every road junction when there is multiple inbound traffic coming to that junction. Within that group, there needs to be a traffic light for each inbound traffic.

For instance, in the below scenario, two traffic light groups are needed, one for the center junction and another for the junction on the right. The first group needs 3 traffic lights since there are 3 inbound traffic directions coming into that junction. The right junction, on the other hand, only needs two traffic lights.

In each group, there has to be exactly one traffic light in green state, while the other lights must be red. Therefore, traffic is allowed for only one inbound traffic for a particular junction.

Traffic lights have two states only: red and green (no yellow state). They face the incoming traffic and are placed on the right side of the road, so you need to set each light's position and orientation correctly.



Each group has an associated timer (the time attribute) and a period (the duration attribute) which determine when the next light needs go green within that group. The timer increments by the timestep attribute every time simulate method is called. When the timer reaches the duration value, the next light in the group goes green while the others in the group go red and the timer resets back to zero. Note that each traffic light group needs to be simulated individually.

The cars should have the same cruising speed but should not crash into each other. So, they may need to stop occasionally. You are allowed to make any modifications you think necessary in Vehicle or Waypoint class to ensure this.

If you can implement smooth acceleration and deceleration of the vehicles at the traffic lights or when there is a car in front, you will get extra credit for the project.

5. Project Deliverables

Your submission should include a zip file containing a project report and the contents of your project folder.

1. You need to add comments to your code. Uncommented code will get partial credit. Be reasonable with the number of comments you add. Do not try to comment every line.
2. DO NOT submit files individually. Put them into a compressed zip archive and submit the zip archive. Name your zip archive with full name of your team members such as ahmet_bilir_and_veli_yapar.zip
3. Submit your homework zip file as an attachment to Ninova.

Below is the rubric for the project report.

Introduction

Briefly describe the project goals here

Team Members

Name the project team members and specify their roles in the implementation of the project.

Implementation

Here, you should describe the classes you implemented, how you designed the traffic lights and groups and their functionality, and if you went for the extra credit, how you handled acceleration and deceleration of the vehicles.

Discussion

Briefly describe the problems you faced in the implementation of your project and how you could improve it.