BLG252E Object Oriented Programming Project 3

1. Objective

The objective of this project is to enhance the traffic simulator you implemented in previous parts.

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 previous parts.

3. Project Description

The project will build upon the previous part (see previous project description for more information). In this final part, you will run the simulation with **at least 6 cars and at least 2 buses**. The vehicle class will be modified as below.

3.1 Classes

In this final part of the project, you should convert Vehicle class to an abstract class with move method being a pure virtual function that takes no parameters:

virtual void move() = 0;

In addition to the classes implemented in Project 1 and 2, you should implement the following classes:

1. **Car**

   This class should derive from the Vehicle class with the move method implementing a car movement based on the previous project rules (following waypoints, deciding on which way to go next randomly at intersections and avoiding collusions with other cars). This means the decision to select the next waypoint must be implemented in the move method.

2. **Bus**

   This class should derive from Vehicle class with the following additional members

   Private Members:

   - **vector<int> stops**; //is a vector of BusStop indexes that the bus will visit in order. When the bus reaches the last stop, it should go back to the first element in the vector container and should repeat its visits starting from the first bus stop. There should be **at least 5 bus stops** for a bus**.** Bus stops can be common between the two busses.
   - **int currentStop**; //current stop index

   Public Methods:

   addStop(int index)

   Adds a BusStop index for the bus to stop at

   This class should also implement the base class move method with one difference that the move method should not pick a path randomly at junctions but should take a path to reach the next bus stop.

3. **BusStop**

Private Members:

- **float x**; //x coordinate of the bus stop
- **float y**; //y coordinate of the bus stop
- **float dir**; //direction of the bus stop (determines the orientation of the bus stop on the map)
- **sf::Texture texture**; //texture for the bus stop
- **sf::Sprite sprite**; //sprite for the bus stop

Public Methods:

BusStop(float x, float y, float dir)

Constructor for the BusStop class

parameters:

- **x**: x coordinate of the bus stop
- **y**: y coordinate of the bus stop
- **dir**: bus stop direction, i.e., orientation

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

Returns the position and the direction of the bus stop

parameters:

- **x**: x coordinate of the bus stop
- **y**: y coordinate of the bus stop
- **dir**: bus stop direction, i.e., orientation

void draw()

Draws the bus stop to screen

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 and how you designed the modified move method.

Discussion

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