***Jamin Tanaskovic CP 2406***

# Console Traffic Simulation

Table of Contents

[*Console Traffic Simulation* 1](#_Toc24120349)

[Problem specification 2](#_Toc24120350)

[How will the program behave 2](#_Toc24120351)

[***Shape <<Abstract>>*** 2](#_Toc24120352)

[***Vehicle <<Abstract>>*** 2](#_Toc24120353)

[***Car*** 2](#_Toc24120354)

[***Bus*** 2](#_Toc24120355)

[***Bike*** 2](#_Toc24120356)

[***Road*** 3](#_Toc24120357)

[***TrafficLight*** 3](#_Toc24120358)

[***Controller*** 3](#_Toc24120359)

[***MainFrame*** 3](#_Toc24120360)

[***GamePanel*** 3](#_Toc24120361)

[***VehicleType*** 3](#_Toc24120362)

[***Abstract*** 3](#_Toc24120363)

[***G.U.I.*** 3](#_Toc24120364)

[***ActionListeners*** 3](#_Toc24120365)

[Resources Used 4](#_Toc24120366)

Problem specification

**What exactly is the problem?**

The problem attempting to be solved will be for a traffic simulation. This problem will be judged on two parts. Part 2 is being tested which is displaying the traffic simulation with Graphical User Implementation. We are to try to use elements from the previous work to make a more fully developed program.

**How will the program be used?**

The programme is meant to be used to determine issues with the placement of Traffic Lights on a street map created by a user. It meant to identify areas of congestion using three types of vehicles Cars, Busses and Motorbikes. All vehicles are meant to follow Australian road rules. The user is to start the programme and either create a new city, load a previous made city, edit a previous made city, or save a created or edited city. They may also set the time rate for the simulation, set the amount of vehicles to be tested, start the sim or stop it.

## How will the program behave

**Problem decomposition using UML class diagrams:** What objects will be used and how will they interact?

### ***Shape <<Abstract>>***

The *Shape* class is an [*abstract*](#_Abstract) class that all objects are derived from, In particular the *Shape* class is extended by the [*Vehicle*](#_Vehicle_<<Abstract>>) class which is also abstract, the [*Road*](#_Road) class, and the [*TrafficLight*](#_TrafficLight) class. This class is used to establish variables used by the aforementioned classes and the classes derived from [*Vehicle*](#_Vehicle_<<Abstract>>).

### ***Vehicle <<Abstract>>***

The *Vehicle* class is also an [*abstract*](#_Abstract) class and extends from the [*Shape*](#_Shape_<<Abstract>>) class bringing into itself the variables and methods created in the [*Shape*](#_Shape_<<Abstract>>) class. In addition to this it adds features that will be used in the child classes [*Car*](#_Car), [*Bus*](#_Bus), and [*Bike*](#_Bike). These three classes extend into themselves both the variables and methods created in *Vehicle* as well as those created in [*Shape*](#_Shape_<<Abstract>>).

### ***Car***

The *Car* class was the first of the three classes created through extending the [*Vehicle*](#_Vehicle_<<Abstract>>) class as explained previously. This class also had to implement the [*abstract*](#_Abstract)methods that had not been previously instantialized. This class is to make a *Car* object to be used by the [*GamePane*](#_GamePane).

### ***Bus***

The *Bus* class was the second of the three classes created through extending the [*Vehicle*](#_Vehicle_<<Abstract>>) class as explained previously. This class also had to implement the [*abstract*](#_Abstract) methods that had not been previously instantialized. This class is to make a *Bus* object to be used by the [*GamePane*](#_GamePane).

### ***Bike***

The *Bike* class was the third of the three classes created through extending the [*Vehicle*](#_Vehicle_<<Abstract>>) class as explained previously. This class also had to implement the [*abstract*](#_Abstract) methods that had not been previously instantialized. This class is to make a *Bike* object to be used by the [*GamePane*](#_GamePane).

### ***Road***

The *Road* class was second of the three classes that extended themselves from [*Shape*](#_Shape_<<Abstract>>)*.* This class added the variable and methods to create the *Road* object. This class is to be used by the[*Controller*](#_Controller) class to create maps, the vehicles to drive upon, the [*GamePane*](#_GamePane) class to be displayed and finally by the [*TrafficLight*](#_TrafficLight) class to add the lights. The Road object was to have two lanes and a minimum length of 12 that of two bus lengths and the maximum length of 30

### ***TrafficLight***

The *TrafficLight* class was third of the three classes that extended themselves from [*Shape*](#_Shape_<<Abstract>>)*.* This class added the variable and methods to create the *TrafficLight* object. The lights can be placed at either the beginning or the end of a lane. The *TrafficLight* class will also change colour from green to red at a preordained rate. The [*Road*](#_Road) class and the [*Controller*](#_Controller) class will use the *TrafficLight* class.

### ***Controller***

The *Controller* class is the class that controls how the program operates. This class is the main class controlling the logic. The *Controller* class responds to user choices made within the [*Mainframe’s*](#_MainFrame) display. It holds the logic allowing you to display the creation of the city, and the [*GamePanel*](#_GamePanel)

### ***MainFrame***

The *MainFrame* class holds most of the Graphical user Implementation ([*GUI*](#_G.U.I.)). This class is used for the coding of the look and feel of the project. This is what the user see’s when the program is used and any selection made by the user send requests through [*ActionListeners*](#_ActionListeners) to the [*Controller*](#_Controller) class.

### ***GamePanel***

The *GamePanel* class was created to be used as a screen that was displayed in the [*MainFrame*](#_MainFrame_1)*.* The *GamePanel* was to use saved new game or edit game and display it in the centre panel. This was to display the actual sim showing the [*Car*](#_Car), [*Bus*](#_Bus), [*Bike*](#_Bike)*,* and created [*Road*](#_Road) map. The [*Vehicles*](#_Vehicle_<<Abstract>>) were to move and interact with [*TrafficLight’s*](#_TrafficLight) and corners. This was not successfully implemented in this project

### ***VehicleType***

The VehicleType class was a helper class. This class was basically created to make randomised types of vehicles to be called for the [*GamePanel*](#_GamePane)

### ***Abstract***

*Abstract* class is a class that cannot be made into an object. An abstract class has one or more unformed methods that need to be implemented by child classes. In this program the [*Shape*](#_Shape_<<Abstract>>) and [*Vehicle*](#_Vehicle_<<Abstract>>) class are abstract. The [*Shape*](#_Shape_<<Abstract>>) class is the base class for building all object that in this case have a rectangular shape, this class also implements methods that can be used by child classes and has a few abstract methods. The [*Vehicle*](#_Vehicle_<<Abstract>>) class adds the methods and variables from [*Shape*](#_Shape_<<Abstract>>) and creates variable and methods that will be used by vehicles. This class is also abstract and [*Vehicle’s*](#_Vehicle_<<Abstract>>) child classes need to implement its abstract methods

### ***G.U.I.***

The graphical user interface is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicator such as primary notation, instead of text-based user interfaces, typed command labels or text navigation. <https://en.wikipedia.org/wiki/Graphical_user_interface>

### ***ActionListeners***

A button listener must implement the ActionListener interface. ActionListener is an interface (not a class) that contains a single method: ... A class that implements the interface must contain an actionPerformed() method. The ActionEvent parameter is an Event object that represents an event (a button click). <https://chortle.ccsu.edu/Java5/Notes/chap57/ch57_10.html>

## Resources Used



