

KING'S COLLEGE LONDON

TRAFFIC SIMULATOR

GROUP PROJECT

---

## Team Diversity

---

*Members:*

Balasz Kiss

Eddy Mukasa

Gabb Visessmit

Pongsakorn N. Riyamongkol

Snorri Hannesson



January 29, 2015

# 1 Introduction

This is the initial report for the group project in King's College London. The project is about making a traffic simulator. This report will have two parts:

- 1 Set out the teams aims for the project, a strategy for achieving those aims, an initial timetable and progress so far.
- 2 How will the team work together: set out the roles each team members is expected to play and how will the team collaborate. How the peer assessment is expected to be handled and how will conflicts be resolved (including, but not limited to, the peer assessment).

## 2 Part one

The simulator will be written in Java and will have a graphical user interface that will be constructed with help of JavaFX. The graphics will be simple as well as the road system. There will be two different types of vehicles, cars and buses. A strategy for emergency service support will be considered and two different traffic management policies will be compared.

### 2.1 Aims for the project

#### 2.1.1 Vehicle types

To make the simulation more realistic the model will have two types of vehicles, cars and buses. Furthermore, each vehicle can be either *cautious*, *normal* or *reckless*. There are two differences between cars and buses. One is that cars go in general faster and have higher acceleration. Another difference is that the buses are larger and therefore take more space on the road, so fewer buses than cars are needed for the system to detect a congestion.

Type	Car (X pixels)	Bus (2X pixels)
Cautious	Acceleration: Y Top Speed: X	Acceleration: Y - 10 Top Speed: X - 20
Normal	Acceleration: Y + 10 Top Speed: X + 20	Acceleration: Y Top Speed: X
Reckless	Acceleration: Y + 20 Top Speed: X + 40	Acceleration: Y + 10 Top Speed: X + 20

A vehicle that is stuck behind a slower vehicle is prone to change lane. The left is the main lane.

### 2.1.2 Emergency strategy

The emergency strategy in the simulator is to prioritise ambulances at traffic lights and on the road. When a ambulance is generated the vehicles that are on the same road as the ambulance will change to the left lane if possible. Other vehicles will not cross the junction or intersection in order to let the ambulance clearly cross the traffic lights.

### 2.1.3 Traffic management

We will be comparing two different traffic management policies:

- Fixed time policy
- Congestion control policy

With the **Fixed time policy** we have will have peak hours (when the traffic density is the highest) and off-peak hours (when the traffic density is normal). On off-peak hours the duration of the green light will be X seconds before it changes to red. On peak hours the duration of the green light will be 2X seconds before it changes to red.

The **Congestion control policy** will always have the duration of green light in each direction unless the system detects a congestion. When a congestion is detected in some direction the duration of the green light will be increased in that direction for one time to avoid traffic jam.

The two different policies will be compared by average time each vehicle is in the system. Each vehicle will have a timer that starts when it enters the system and gets written to a log when the vehicle exits the system. The average time that a vehicle is in the system is calculated and the the policy that generates lower average time is considered better.

## 2.2 Strategy

*I don't know what will come here. Maybe Uml diagram.*

### 2.2.1 Initial timetable

*Some use cases and stuff and time of each iteration and which use case we should be done with. Maybe the first iteration will end the day we hand in this report.*

## 2.3 Progress so far

*Could have burndown graph where we have finished the first iteration*

## **3 Part two**

### **3.1 Project management**

*Project management style based on agile. Based on iteration, try to come closer to final product with each iteration.*

### **3.2 Roles**

Snorri: Coordinator, Tester Gabb: Graphical gurus. Balasz: Lead programmer. Eddy: Architect Neab: ??

### **3.3 Collaboration**

*We use facebook groups, github so far. Perhaps we should look into using wikis on github for info or trello for project management and use cases.*

### **3.4 Peer assessment**

*Every meeting is logged and we see on github who did what so we will have some idea on what each member contributed during the project.*

### **3.5 Conflicts**

*Arm wrestle???*