Part I
Project Description

1 Introduction

The problems related to traffic and transportation intensify greatly as towns and cities develop and grow. The field of Traffic Engineering aims to plan transportation with reasonable precision for the sake of avoiding costly, and often irreversible, mistakes. Through an application of basic knowledge and theories in the field, road systems can be developed effectively where a trial and error approach is simply impractical.

Only by systematic evaluation can we upgrade our systems and discard ineffective designs. A direct approach to evaluating the design possibilities in transportation systems is by the use of computer software that simulates an instance of the real world. Traffic Simulation is the art of mathematically modelling such systems using software to help better the planning, design and management of them.

2 Aims

The aim of this project is to develop Traffic Simulation software as a team that can effectively fulfil these goals. We will be researching, planning, designing and implementing a software product capable of simulating traffic whilst paying close attention to group collaboration and work ethic.

We developed a hierarchy of aims that highlight the level of importance of each goal by using MoSCoW analysis. Naturally, aims with a higher level of importance should be wholly completed before work towards the other goals has begun. In this way, the chance of producing a complete and well-functioning software artefact is maximised. The produced aims are listed below by order of importance:

2.1 Must

1. The system must simulate individual vehicles operating in different parts of a road network.
2. The system must deal with vehicle entrance and exit points in a way that simulates a real world environment.
3. The system must allow for the individual behaviour of vehicles.
4. The system must allow different traffic management policies to be plugged in and compared.

5. The system must compute statistics of interest such as Vehicle Miles Travelled (VMT), mean system speed, total system delay, etc.

2.2 Should

1. The system should have support for emergency services.
2. The system should allow the user to create the map in which to simulate traffic.

2.3 Could

1. The system could save and load maps from a file.
2. The system could use data from an external source such as Google Maps to generate the area in which to simulate traffic.

2.4 Won't

1. The system will not be rely on any existing traffic simulation platforms, engines or modelling frameworks. We restrict our efforts to the use of APIs of our development language and toolset.

3 Progress

Although the planning phase took longer than anticipated, ultimately it has been a fruitful exercise with little deviation from the project milestones. We have successfully selected a microscopic discrete simulation model for it's relative simplicity. Using a Model-View-Controller(MVC) design pattern a prototype has been developed with a working visualiser. At present we have met the first two requirements of 2.1 by representing, in the model and visualiser, numerous intersections with opposing roads, each containing two lanes. Vehicles are spawned from designated entrance points once another leaves from the exits. Vehicle flow is successfully controlled by traffic lights and basic policy management functionality, such as the number of cars or simulation tickrate, is operational.

The feedback from this discrete prototype has lead to us entertaining a continuous model approach. Although the discrete method has the added advantage of being efficient for larger map representations, further research has shown we can achieve a more realistic vehicle behaviour using a continuous space model in combination with discrete time.

Date	**Phase**
15/01/15	Formation & Research
23/01/15	Research
30/01/15	Development Cycle 1
06/02/15	Development Cycle 2
13/02/15	Development Cycle 3
20/02/15	Development Cycle 4
27/02/15	Report

Table 1: Timetable

Part II
Project Organisation

4 Working Policy

From the onset it was established we will be adopting a cooperative and largely democratic approach to team management. Our determination to attain the maximum results from our proposed effort entitles each team member equal input. With the equal partnership each member is aware their commitment is directly related to the success of the team and thus we felt a policy encouraging self-organisation more suitable than a top-down hierarchy. However an individual was nominated to be Project Coordinator thus acting as a single point of contact for the team and the Informatics Department.

5 Development Methodology

Considering the particular challenges we presumed to encounter, specifically time-management, the brief development window and minimal exposure to working in teams, the 'Iterative and Incremental' development approach was favoured. In general this technique involves the increase in feature additions and a cyclical release pattern with great emphasis on feedback. The evaluation at the end of iterative cycles allows for continued planning, analysis and design producing an incremental addition in features.

6 Project Roles

Having identified the major phases of the project and critical tasks, we agreed the following role definitions:

1. **Research & Designer** - responsible for delving into the literature and suggesting solutions to our challenges.
2. **Developer (Graphics & GUI)** - responsible for translating functional graphical requirements into deliverables.
3. **2 x Developer (Model Implementation)** - responsible for translating functional requirements of the model and back-end into deliverables.
4. **Documentation Analyst** - responsible for documenting the system implementable, the current status of the project and report writing.

It is paramount to note that we are not ultimately rigid in our role selection. Individuals may take on the responsibility, possibly more than one, of each of the defined roles for a set period. Team members are then able to experience the challenges faced from all perspectives and has the added advantage of flexibility i.e. to re-assign an individual should they be required elsewhere.

7 Collaboration

We have found GitHub to be exceptionally useful in managing collaborative effort, not only for the developers generating code but also those producing research, design or planning output. At the start of each iteration cycle we are translating identified tasks from the project plan into issuable GitHub tasks that can be assigned to individual(s) to progress. Furthermore GitHub has an excellent wiki feature where our contributors can document the many facets of the project.
In addition we are effectively employing a Facebook group for open discussion and operational issues such as absence. Evernote, a multi-platform collaboration system, is being utilised for meeting minutes and hosting large files. Finally the group has found the use of instant messaging application WhatsApp beneficial with time sensitive notifications.

Although the state of collaborative platforms are feature-rich we have found face-to-face meetings pivotal in our efforts. Thus we have scheduled as a minimum one major and minor project meeting a week.

8 Peer Assessment & Conflict Resolution

As a direct result of our working policy encouraging a proactive mindset, we have opted to award all team members equal weighting of the final result. At the point prior to submission we will meet to ratify equal contribution and if there are any concerns, a majority vote will be issued. This majority vote strategy is being used for wider issues of contention. Conflicts are resolved using the following rules:

1. Each individual has the opportunity to voice their concerns.
2. Each individual has an equal vote on an issue.
3. The majority vote is the deciding vote.

The largest point of contention is likely to be the completion of tasks or lack of by team members. Therefore clearly defining roles and responsibilities from the onset is our identified method to ensure more favourable team dynamics.