Traffic Simulator

KeepItClean

February 2016

1 Project Description

1.1 Project Aim

Project Background

Traffic simulation software plays an important role in defining the effective traffic management policies. A traffic policy must be simulated and verified before it is actually implemented in real life. Otherwise, it might harm road users safety and traffic efficiency.There are a lot of factors contributing to road accidents and traffic efficiency. The major factors include speed, traffic light timing and drivers behaviours.

Therefore, this project will simulate traffic management policies focusing on those 3 main factors ie. Speed limit; Traffic light timing; Drivers behaviours.

The software will test different policies relating to these factors and analyse the traffic efficiency and safety in relation to traffic density, enforcement policy and drivers behaviours.

The analysis will be based on the following metrics:

- Probability of crashes (%): percentage of crashes in total vehicles per session. This metric is to measure the safety of the policy
- Average speed: average speed of vehicles in a session. This metric is to measure traffic efficiency and reliability.

Project Scope

The software is to simulate the traffic management policy following UK highway code which is left-lane oriented and using the speed limit within the range of speed limit defined in the UK highway code.

Within the constraints of time and resources, the projects scope includes:

- Controlled Map: Minimum multiple-lane roads and a junction. It should support the scalability to a complex traffic network (ie. Multiple roundabouts, multiple junctions, multiple traffic lights, bus lane).
- Vehicles: Simulate multiple types of vehicles, which include at least cars and ambulances (emergency vehicle) and at least three classes of drivers behaviours (reckless, cautious, normal).
- Policy: the project must support at least fixed control policy (traffic light timing, speed limit). It should be scalable to variable control policy.
- Simulation engine: must test the policies with different levels of traffic density.
- Report: Must provide the statistics and calculates metrics as above.

1.2 Project Approach

Management Approach: Scrum The team decided that due the nature of the project it was needed an strategy that allowed progress every predetermine period of time and, at the same time, to have something ready to deliver. For this reason the decision was to follow an Agile development approach using Scrum in order to have a management appropriate for a small development time and have a specific artefact ready to go at the end of each iteration.

The approach we are going to follow is the Scrum methodology focused in the team goals and needs. This means that even when people will be assigned to an specific role, it does not mean that she or he will be only do tasks related. The tasks will be balanced and assigned according to the Sprint number, task complexity, and task completion percentage. It also means that some features of the Scrum development methodology will be modified accordingly and only task progress will be reported.

Taking in count the available development weeks, the Sprint length was defined to last two weeks. At the beginning of each sprint a Scrum Planning Meeting will be held. This is where the team will decide what will be the next target deliverable for the sprint. The daily stand up meeting will be replaced by mechanisms explained in the Communication Process subsection. Next the role assignation is shown. Scrum Master: Daniel Mendoza. Product Owner: All the team will be involved. Development team: All the team will be involved.

Technical Approach: Java The team made the decision to develop in Java SE because it was the technology whom all team members have at least some experience.

The next thing that motivate us to chose this technology was the necessity to reduce complexity as much as possible from the beginning due a tight development schedule.

By using Java we are removing the need of have a manual memory management in order to avoid memory leaks.

Java also give us a way to easily deploy the software in many platforms without having special implementations for each one. This reduces development time and removes the platform concerns.

Quality Management: Unit tests and Test Apps Quality of the software will be tracked during every iteration. The strategy we are going to follow is create unit tests for each subsystem, as well as testing applications when necessary.

To develop unit test we are going to use the JUnit framework. It is a widely used way to develop unit test, therefore it will be easy to investigate implementation strategies for our project.

Each sprint will have two test tasks by default: Integration Test and Regression Test. The integration test will consist in test the subsystems interactions that changed after the development work has being completed. The regression test will consist on a set suite of predefined minimal features of the system that have to be executed in order to make sure previous functionality works as expected.

1.3 Project Schedule

For this application, we have total 4 iterations after the Intermediate Report (1st Report). Detail for milestone can be seen in Table 1. The detail for iterations and the project functionality can be seen in Table 2.

1.4 Initial Progress

Our works so far is the design of the application, including the UML for classes and objects that we need.

Table 1: Project Milestones

Milestone	Deliverable	Date	Dependant Upon
1st Report	Intermediate Report complete (Iteration 0)	Feb 9	Intermediate Planning and Initial Progress
Initial	Application Ready (Iteration 1)	Feb 18	First iteration of bugs fixed. Iteration 1 features implemented. Program ready for unit test.
Mid	Application Ready (Iteration 2)	Mar 3	Second iteration of bugs fixed. Iteration 2 features implemented. Program ready for unit test.
Pre-Final	Application ready (Iteration 3)	Mar 17	All features and bugs fixed. Program finalised.
Final	Report complete and application ready (Iteration 4)	Mar 31	Completion of packaging and program. Final report ready. All testing done.

2 Project Organisation

2.1 Roles

Team members will be assigned to an specific role, however this does not mean that she or he will be only doing tasks related. This ones will be balanced and assigned according to the Sprint goal, task complexity, and task completion percentage.

- Scrum Master: Daniel Mendoza.
- Product Owner: All the team will be involved.
- Development team: All the team will be involved.

2.2 Collaboration Tools

- Github
- Trello

2.3 Process Handling Peer Assessment

criteria: punctual, communicated, proactive, functionally sufficient.

2.4 Communication Process

- Meeting: We replaced the daily scrum meetings by having two weekly meetings Mondays and Thursdays.
- Tools: Trello to keep track of tasks and share progress. Whatsapp group.
- Conflict Handling

Table 2: Project Iteration

Iteration0	Iteration1	Iteration2	Iteration3	Iteration4
Requirements	FT1:Map-Lane	FT5:Map-Roundabout	FT7-2:Map-Complex	Final Report
System Architecture Design	FT2:Map-Junction	FT7-1:Map-Multiple Roundabout	FT-10:Vehicle-Buses	Final Bug Fixing
Component Detailed Design	FT3:Map-Traffic Light	FT14:Vehicle-Driver's Behaviour	FT17:Policy-VariableControl	Final Testing
Initial Report	FT4:Map-Road	FT15:Vehicle-Emergency	FT22-2:GUI-consolidated added vehicles/maps and animation	
	FT6:Map-Network Boundary	FT20:GUI-Roundabout	FT23-2:Optimal Simulation Engine	
	FT18:GUI-Lane	FT21:GUI-Consolidated and Animation		
	FT19:GUI-Junctions Traffic Light	FT24:Data log and analysis		
	FT21:GUI-Multiple Vehicles			
	FT8:Vehicle-Generic			
	FT9:Vehicle-Car			
	FT12:Vehicle-Acceleration Deacceleration			
	FT11:Vehicle-Traffic Light			
	FT13:Vehicle-Change Lane Direction			
	FT23:Simulation Engine			
	FT16:Policy-Fixed Control Policy			