NEA: Traffic Simulator

Joshua Smart

June 2021 - June 2022

Contents

1 Analysis **2**
1.1 Project Overview . . . 2
1.2 The Problem . . . 2
1.2.1 End user . . . 3
1.3 Current Systems . . . 3
1.3.1 AnyLogic - Road Traffic Simulation Software . . . 3
1.3.2 SOUND . . . 3
1.3.3 Eclipse SUMO . . . 3
1.4 Research . . . 3
1.4.1 Traffic laws . . . 3
1.4.2 Graphs . . . 3
1.5 Proposed Solution . . . 5
1.6 Objectives . . . 5

2 Documented Design **6**

3 Technical Solution **7**

4 Testing **8**

5 Evaluation **9**

References **10**

1 Analysis

1.1 Project Overview

Physical infrastructure is a large and complicated subject, encompassing everything from running water to mains electricity. Below I have listed the main features included in physical infrastructure:

- **Transportation:** Road and highway networks; Mass transit systems; Railways; Canals; Seaports; Airports; Bicycle paths / pedestrian walkways
- **Energy:** Electrical power network; Natural gas pipelines; Petroleum pipelines; Coal production and processing
- **Water management:** Drinking water supply; Sewage collection; Drainage systems; Irrigation systems; Flood control systems; Coastal management
- **Communications:** Postal service; Telephone networks; Mobile phone networks; Television and radio stations; Internet services; Communications satellites; Undersea cables
- **Solid waste management:** Landfills; Incinerators; Hazardous waste disposal

Computer modelling systems could be developed for any one of these areas, allowing users to prototype infrastructure designs before the costly process of constructing it. For this project I have chosen to focus on the Transportation sector, as it can include some of the most expensive forms of infrastructure.

Naturally the major responsibility of transportation infrastructure goes towards the construction and maintenance of road networks, including but not limited to junctions, roundabouts, highways and traffic lights. These networks can get very complicated and difficult to manage, for example the UK's so called "spaghetti junction" in Birmingham.

Figure 1.1: Birmingham's "Spaghetti junction" [1]

1.2 The Problem

Development of transportation is a very expensive and time consuming process, so being able to evaluate the efficiency and cost-effectiveness of road layouts beforehand would be very beneficial. This project aims

to develop a road network simulator that can be used to evaluate the efficiency of inputted designs under a range of different traffic conditions.

1.2.1 End user

1.3 Current Systems

1.3.1 AnyLogic - Road Traffic Simulation Software

AnyLogic - Road Traffic Simulation Software [2] is an industry-level program used for analysing traffic patterns and behaviours, below are a couple images of the program.

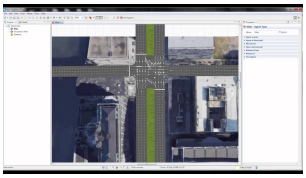

Figure 1.2: Happy Smiley

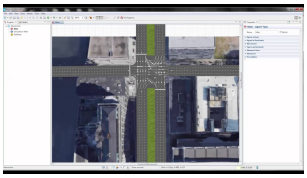

Figure 1.3: Sad Smiley

Figure 1.4: Sad Smiley

1.3.2 SOUND

1.3.3 Eclipse SUMO

1.4 Research

1.4.1 Traffic laws

This project will be based off the UK Highway Code [3] as of the latest update (23 March 2021). This document contains all laws and regulations for driving in the UK. The relevant points include:

- Normal driving position is considered the leftmost lane of the road and should be assumed whenever possible
- Right-of-way is given to the major road when emerging from a junction
- When entering a roundabout, you must give way to vehicles on your right
- Is it recommended to maintain a two second separation distance from the vehicle in front

These rules will be used to inform how the vehicles in the simulation operate in different circumstances. Although it may be noted that not all vehicles follow these laws strictly, so deviations may be added to account for this.

1.4.2 Graphs

The natural way to represent any kind of network (including road networks) programmatically is using a graph, so I will conduct some preliminary research into this topic.

Graphs are an abstract data structure used to describe a set of vertices and the edges connecting them, both vertices and edges can have associated values. This is a very useful structure in computer science as it can be used to model a wide range of existing data such as power grids and social networks. Graphs also come with a wide range of existing algorithms for operating on them such as Dijkstra's shortest path algorithm or the A* search algorithm.

Described below are the two most common data structures used to describe graphs:

- **Adjacency list** - Vertices are stored as objects containing a list of it's own adjacent vertices.
- **Adjacency matrix** - A two-dimensional matrix where each cell represents the edge (or lack of edge) from the vertex described by the row to the vertex described by the column.

Each implementation has its advantages and disadvantages Table 1.1 shows the time complexity of each operation over these two structures.

	Adjacency list	Adjacency matrix
Store graph	$O(\|V\| + \|E\|)$	$O(\|V\|^2)$
Add vertex	$O(1)$	$O(\|V\|^2)$
Add edge	$O(1)$	$O(1)$
Remove vertex	$O(\|E\|)$	$O(\|V\|^2)$
Remove edge	$O(\|V\|)$	$O(1)$
Check for adjacency between two vertices	$O(\|V\|)$	$O(1)$

Table 1.1: Time complexities of different graph implementations [4]

Since road networks are generally optimised for maximum throughput I will also research the graph maximum flow problem as this may be useful in evaluating the networks. Formally, the maximum flow problem is as follows [**citation needed**]:

- Let $N = (V, E)$ be a network with $s, t \in V$, where s is the source and t is the sink.
- The capacity of an edge is the maximum flow that can pass through it, the capacity from vertex v to vertex u is denoted as c_{uv}; and flow on the same edge as f_{uv}.
- The flow network must satisfy two properties:
  - Capacity constraint: The flow of an edge cannot exceed it's capacity.

$$\forall (u, v) \in E : f_{uv} \leq c_{uv}$$

  - Conservation constraint: The flow into a vertex must equal the flow exiting a vertex (the source and sink are exceptions).

$$\forall v \in V \setminus \{s, t\} : \sum_{u:(u,v)\in E} f_{uv} = \sum_{u:(v,u)\in E} f_{vu}$$

- The value of flow is the total amount of flow passing from the source to the sink, given by:

$$|f| = \sum_{v:(s,v)\in E} f_{sv}$$

The goal of the maximum flow problem is to route as much flow as possible from the source to the sink under these constraints. Many algorithms to compute $f_{\max}$ have been found including but not limited to Linear Programming, the Ford-Fulkerson algorithm or the Edmonds-Karp algorithm.

The maximum flow of a road network could be a good indicator of the theoretical maximum efficiency.

1.5 Proposed Solution

My proposed solution consists of a desktop app that allows the user to construct an arbitrary road network and test it using a set of 'car' agents that traverse the network.

1.6 Objectives

2 | Documented Design

3 | Technical Solution

4 | Testing

5 Evaluation

Bibliography

[1] Wikipedia, the free encyclopedia. *Highways Agency photo on flickr*. Own work derived from File:Aerial view of M6A38 Spaghetti Junction.jpg, whose license permits such derivative work when properly credited, CC BY 2.0. 2008. URL: https://commons.wikimedia.org/w/index.php?curid=19927460.

[2] *Road traffic simulation software*. URL: https://www.anylogic.com/road-traffic/.

[3] *The Highway Code - Guidance - GOV.UK*. URL: https://www.gov.uk/guidance/the-highway-code.

[4] *Time and Space Complexity of Adjacency Matrix and List — Baeldung on Computer Science*. URL: https://www.baeldung.com/cs/adjacency-matrix-list-complexity.