Project plan

Martin Blom Felix Jönsson Hannes Kaulio Arvid Nyberg
Marcus Schagerberg Jakob Windt

2023-01-23

Bachelors Thesis
Computer engineering and IT
Chalmers University of Technology

Contents

Glossary

Unity, Crossplatform game engine.

Prefab, A reusable object in Unity that stores a configuration and can be used as a template for creating assets.

Agent, Autonomous systems that inhabits an environment and act based on predefined rules.

Scrum, The scrum agile project management framework provides structure and management of work and is popular among software development teams.

1 Introduction

Traffic congestion is the result of the demand for road/ -and railway travel exceeding the supply. This problem can be seen all around the world[1] and it impacts our quality of life. As vehicular traffic builds up, bikes, cars, busses and trams can become stationary or move significantly slower, wasting both time and fuel. Moreover, delaying transportation of goods can lead to huge economical costs, food waste and general inconvenience for effected parties.

Not only does this affect our society on the larger scale, but also on an individual scale. The average citizen will spend around a year of their life commuting if the average commuting time doesn't go down. According to Trafikanalys data of traffic habits[2], the average Swedish citizen's daily commuting time during 2019 was just under 1 hour and dropped to around 45 minutes post-COVID. Some amount of commuting time is inevitable in our current society. But if you look at some of the bigger cities in the world such as London, an estimate of 156 hours per person was lost in just traffic delay alone during 2022[1].

Other than leading to loss of time and resources, congestion and traffic in general leads to air pollution which poses health hazards and also lowers the quality of life[3]. A study of air pollution in connection to cars made in the USA during 2022, shows that transportation stood for 27% of the total greenhouse gas emissions[4]. By reducing the amount of combustion sources that contribute to air pollution we can work on solving multiple problems at once.

In section 2 we will present how we aim to create a tool to help with both understanding why, and solving societal, economical and resource problems related to traffic and congestion.

2 Purpose

The purpose of this project is to create a traffic simulation tool that will allow the user to design a road-network, and see real-time statistics about its traffic flow and environmental impact. In addition, the user should be able to manipulate different aspects of the simulation such as the amount of cars.

By changing some of the aspects in the simulation, the user will be able to see whether their design improves a road-network, which can result in less traffic congestion and a smaller environmental impact.

3 Problem

In order to achieve a traffic simulation that clearly and visually shows environmental impact as well as traffic congestion, several problems will have to be solved.

There are many ways to go about when simulating a traffic flow, but based on the purpose and the established limitations of our project we have chosen to implement an agent based simulation which offers a fitting high level of granularity of interplay between the environment and the individual agents.

Firstly, the environment in which all agents respond to and interact within will have to specified and created. Static environmental objects such as roads and corresponding traffic signs will have to be generated in a modular manner to allow the end user to set up and run their customized simulation.

Furthermore, all individual traffic elements such as cars, traffic lights, buses, etc will have to be simulated. In order to accomplish this, agent-based modeling will be used [5]. All these traffic elements will be simulated as individual agents abiding to a set of rules. The problem will be to decide the different rules and logic for the various agents. Vehicle agents will for example need to interact with the different traffic signs and follow the specified road rules. There also exists implicit and more subtle rules that govern movement in public spaces

As a result of using an agent-based approach, performance will surely become an issue. All agents in the simulation will have to be continuously updated according to their rules. This can become computationally expensive when the number of agents increases. Performance-based design choices will have to be made for city-scale simulations to be possible.

Other issues that will have to be resolved is how to process the order of execution both from a low level perspective such as order of execution of functions within an agent or environment object, as well as on a higher perspective such as order priority amongst different entities in the simulation. Failing to do so might otherwise result in subtle faulty behavior that can be hard to track down the cause of. Unity already have some native support for this in the form of different update events with varying priority that Unity objects (classes inheriting from Unity's base class, monobehaviour) can subscribe to in order customize the order of execution. It is however not clear at the moment of writing this paper if this will be enough to ensure easy configuration of execution priority or offer the necessary transparency. Decisions will have to be made if the team should implement our own ABM framework or make use of existing open source unity assets such as ABMU.

Finally, since the goal of our project is to offer a user-friendly tool, a user interface will have to be developed. This interface needs to clearly communicate which parameters the user can tweak in the simulation and offer an intuitive way of doing so. The interface will also need to display relevant information with regards to statistics of the simulation in a manner that is easy to comprehend. Consideration will have to be put into which design patterns to implement to enable this.

4 Limitations

To limit the scope of the simulation tool, it was decided to set some boundaries on what the tool should and should not include.

To begin with, the tool will only simulate vehicles such as cars and buses since including pedestrians was deemed too far fetched because of the added complexity and their small impact on traffic. In addition, there was a worry about the tools overall performance with both vehicles and pedestrians moving around.

Furthermore, there was a discussion about whether the tool would display the map in 2D or 3D. Creating the tool in 2D would take less time to complete, since the overall complexity of the project would decrease. However, by showing the map in only two dimensions; bridges, tunnels, and highway exits/entrances would be harder to interact with compared to in 3D. Also, since the tool is meant to be used by a third party, there is a need to make it appealing. Therefore, the group made the decision to create the tool in 3D, since it allows for more precision when designing road-networks to the user, while also appearing more intuitive.

Lastly, the decision was made to only include a few models of cars with different pollution amounts. These amounts will be similar to Sweden's average emissions by car, but will not be exact since the correct data will be up for the user to input. The reasoning behind this is because the average emission depends on the cities exact location, as well as when the data was collected because of its constant change.

5 Method

The tool will be developed in Unity, a cross platform game engine. In order to collaborate, the project will be stored in a Git repository. This also allows for version control and the opportunity to revert to previous versions for identification of bugs or if something would corrupt. Since a Unity project does not consist of pure text or code files, additional steps have to be taken in order to avoid merge conflicts or other issues that can arise when collaborating in a version control system. Therefore each developer will have their own scene in Unity and updates to the project will mainly take place as changes to prefabs, which are reusable objects that can be used as templates.

5.1 Game Engine and Libraries

Unity is a cross-platform game engine and development environment which comes with a plethora of useful tools for developing software that communicate a 3D environment. It is currently the most used game engine, holding a market share of 38% within the game engine market. From initially being marketed as highly accessible game-engine, it has with time grown into a versatile tool that can be used for a wide range of applications, including animation, simulation, VR, architectural visualisation, and more. With the above in mind, together with the massive amount of high quality documentation available and high number of asset plugins available, we motivate our choice of engine to realise the simulation in.

5.2 Modeling

We motivate our choice of going with an agent based approach by its ability to capture and convey complex behaviour that otherwise would have required very complex mathematics. Since we "only" need to decide and configure how the agents react to the simulation environment, this will typically be enough to allow us to capture interesting emergent system behaviour that would have required the use of complex differential equations if modeled from a macroscopic perspective using techniques derived from fluid dynamics.

Since our goal is to make a product that can easily convey both statistical information of the simulation as well as the first-hand experience of how the traffic actually will behave in the form of exploring the environment in real- time 3D, this further motivates our decision to go with a ABM approach. While ABM is classified as a microscopic model, it still allows for gathering data from each autonomous agent and aggregating this in the form of statistics. On the other hand, by actually rendering the simulation and each agent in real time, this will allow the user to see the effect of the parameters they supplied and set for the environment. This could for example be in the form of observing a crossing, or following one specific agent.

ABM also offers a high degree of flexibility where the model makes it easy to construct and inject new agents with different behavior, heuristics, and so on. This will be important both for the development team and the end user experience since this property will allow us to iteratively adapt the complexity of our simulation as the development process proceed, and then easily offer our users way of tweaking different behaviors. The ability to easily adjust the complexity of the model is highly valued by the development team as this allows us to adapt and make changes as needed during the course of the project. This is particularly important as the team has not previously worked on a project of this nature, and thus, flexibility serves as a safety net for any potential shortcomings in product planning.

5.3 Testing

Due to the challenging nature of validating and testing agent-based simulation, our main method of making sure our software is working correctly will be in the form in-house user testing. A set amount of hours will be allocated each week once testing becomes relevant where developers will explore and observe the simulation under different parameters. The goal of this will be to identify any faulty system behaviour or usability issues, and also gather valuable feedback on the overall design and functionality of the simulation.

5.4 Organisation

In order to have a clear organizational structure we decided to employ an interpretation of the scrum agile framework [6]. In scrum, the work is split into weekly partitions called sprints. Before each sprint there is a planning meeting in which you plan the upcoming sprint - what to do and who should do it. After the sprint, a sprint review is held to reflect about whether all goals were met. The sprints should be relatively short since the scrum framework is based around the idea of creating software through iterations, meaning new iterations should be produced at a regular pace. We have decided on one week long sprints, with a weekly demo meeting acting as our sprint review in which all members demonstrate their completed work during the past sprint. This also serves an additional purpose - sharing the progress between all members so everyone is on board with how far the software as a whole has evolved and opening up possibilities for input and discussion.

After the demo meeting, a sprint planning will occur every week in order to assign tasks to members and plan the goals for the upcoming week. Each weekly sprint planning originates from a goal set for a longer time period which describes the next iteration of the software. The next iteration is collectively decided upon as a specification of requirements during a planning meeting. The planning meeting is centered around the monthly goals set in 6 Time plan. We have also assigned a project leader responsible for taking this next iteration and splitting it up into tasks to be completed during the sprints, making sure there is a logical order of completion for the tasks.

The aim of the combination of longer time frame iteration planning and short term sprint goals is to develop a fast-paced iterative software development environment where new functioning versions of the software are released at a regular interval. This makes sure the software is continuously developed in the right direction and splits the work into small manageable parts with a clear progression between them.

6 Time plan

7 Relevant societal and ethical aspects

Ethical aspects can be broken down into two parts: Aspects related to the method of the project, and possible consequences for users of the final product and society as a whole. There are no obvious ethical issues related to the method of our project. For example, there is no use of personal data, no significant security risks, and no use of questionable technology or software.

The finished product however, brings some interesting aspects to light. One of the goals of this project is to create a tool that can be used by architects, engineers and decision makers working on transport systems by giving them insights about efficiency and emissions. Therefore, we have to look at how our design choices could affect those users in their decisions about transport systems. In regards to this, the most important thing is to be aware of our simulation's limitations as a way to model real life systems.

References

[1] Inrix, "Inrix 2022 global traffic scorecard," https://inrix.com/scorecard/, accessed: 2023-01-30.

[2] "Resvanor," https://www.trafa.se/kommunikationsvanor/RVU-Sverige/, May 2022, accessed: 2023-01-30.

[3] A. J. Cohen, H. R. Anderson, B. Ostro, K. D. Pandey, M. Krzyzanowski, N. Künzli, K. Gutschmidt, C. A. Pope III, I. Romieu, J. M. Samet *et al.*, "Urban air pollution," *Comparative quantification of health risks: global and regional burden of disease attributable to selected major risk factors*, vol. 2, pp. 1353–1433, 2004.

[4] K. Gallagher, "How much air pollution comes from cars?" https://www.treehugger.com/cars-are-causing-air-pollution-we-breathe-new-study-finds-4856825, Aug 2022, accessed: 2023-01-30.

[5] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," https://www.pnas.org/doi/10.1073/pnas.082080899, accessed: 2023-01-24.

[6] C. Drumond, "A guide to scrum: what it is and how it works," https://www.atlassian.com/agile/scrum, accessed: 2023-01-30.

Appendix

A Appendix 1

Figure 1: PLACEHOLDER: Unity logo

B Appendix 2

This is where we will place appendix 2