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.

1 Introduction

Placeholder introduction

Read the MOP book [1]

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 a microscopic 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 [2]. All these traffic elements will be simulated as individual agents all abiding to a set of rules. The problem will be to decide all 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 road rules.

As a result of using an microscopic 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.

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.

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 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.

6 Relevant societal and ethical aspects

As stated above, one of the purposes 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.

References

[1] R. Johansson, *Maskinorienterad programmering med MD407.* Roger Johansson och Göteborgs Mikrovaror, 2020.

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

Appendix

A Appendix 1

Figure 1: Unity logo

B Appendix 2

This is where we will place appendix 2