JOHN VON NEUMANN
FACULTY OF
INFORMATICS

THESIS

OE-NIK
2023

Student’s name: **Djuroska Gergő**
Student’s registration number: **T/006654/FI12904/N**

Óbudai University
John von Neumann Faculty of Informatics
Software Engineering Institute

THESIS WORK SPECIFICATION

Name of the student: **Gergő Djuroska**
Identification number of the student: T/006654/FI12904/N
Neptun's code: [redacted]

Title of the thesis:

Simulation and optimization of smart-node-based traffic control systems in Unity
Okos kereszteződésen alapuló forgalomirányító rendszerek szimulálása és optimalizálása Unity-ben

Internal supervisor: Miklós László Sipos
External supervisor:

Deadline: 15 December 2023

Subjects of the final exam: Computer Architectures
Cloud service technologies and IT security specialization

The task:

The aim of the thesis is to create a 3D simulation in Unity, which visualizes and optimizes traffic at road intersections using smart traffic lights. The system should be able to ensure the minimum waiting time for vehicles passing through the intersection, as well as the optimal utilization of the intersection. To achieve this, a pseudo-artificial intelligence algorithm should be created in C# that controls the decentralized decision-making of the traffic lights based on data received from the appropriate sensors. Investigate which available or to-be-developed technologies can be implemented in the traffic light's sensors. Propose networking solutions for the system's components and support the decision with arguments in favor of the given technology. The solution should include a simplified model of driver behavior. Connect this model with the model of the intersection, which ultimately enables the simulation of different traffic situations. To test the system, evaluate these simulations for system effectiveness. Analyze testing options and prepare the individual test types and test cases. With this solution, it should be possible to build a more complex network (more than one intersection) from smart nodes.

The thesis must contain:

- the exact description and detailed analysis of the task to be solved,
- a high-level structural model of the application,
- possible technologies that can be used to implement certain functions,
- the plan and block diagram of the prepared system,
- the process of implementation, the problems that arise and the solutions given to them,
- the testing plan, testing methodology and testing results,
- the possibilities of using the finished software,
- the presentation and evaluation of the results for the entire work.

ÓBUDAI EGYETEM Neumann János Informatikai Kar 1. Budapest

Place of stamp

…………………………
Dr. Zoltán Vámossy
head of institute

This specification is valid until: **15 December 2025**
(According to OE TVSz 55.§)

I consider the thesis suitable for submission:

………………..
external supervisor

………………………
internal supervisor

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

HALLGATÓI NYILATKOZAT

Neumann János Informatikai Kar

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 2024. december 15.

....

hallgató aláírása

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

KONZULTÁCIÓS NAPLÓ

Hallgató neve: DJUROSKA GERGŐ

Neptun Kód: [redacted]

Tagozat: NAPPALI TAGOZAT

Telefon: Levelezési cím (pl.: lakcím):

[redacted]

Szakdolgozat címe magyarul:

OKOS KERESZTEZŐDÉSEN ALAPULÓ FORGALOMIRÁNYÍTÓ RENDSZEREK SZIMULÁLÁSA ÉS OPTIMALIZÁLÁSA UNITY-BEN

Szakdolgozat címe angolul:

SIMULATION AND OPTIMIZATION OF SMART-NODE-BASED TRAFFIC CONTROL SYSTEMS IN UNITY

Intézményi konzulens: SIPOS MIKLÓS LÁSZLÓ

Külső konzulens: -

Alk.	Dátum	Tartalom	Aláírás
1.	2023.03.06.	Félév teendőinek, menetrendjének és mérföldköveinek ismertetése.	
2.	2023.03.20.	Hasonló rendszerek elemzése.	
3.	2023.04.10.	Irodalomkutatás, annak összegzése valamint konzekvenciák meghatározása.	
4.	2023.05.08.	Tervezés, követelmények specifikációja. Véglegesítés. Leadással kapcsolatos teendők egyeztetése.	

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. tantárgy követelményét teljesítette, beszámolóra bocsátható.

....

Intézményi konzulens

Budapest, 2023. 05. 08.

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

KONZULTÁCIÓS NAPLÓ

Hallgató neve: DJUROSKA GERGŐ

Neptun Kód: [redacted]

Tagozat: NAPPALI

Telefon: Levelezési cím (pl.: lakcím):

[redacted]

Szakdolgozat címe magyarul:

OKOS KERESZTEZŐDÉSEN ALAPULÓ FORGALOMIRÁNYÍTÓ RENDSZEREK SZIMULÁLÁSA ÉS OPTIMALIZÁLÁSA UNITY-BEN

Szakdolgozat címe angolul:

SIMULATION AND OPTIMIZATION OF SMART-NODE-BASED TRAFFIC CONTROL SYSTEMS IN UNITY

Intézményi konzulens: SIPOS MIKLÓS LÁSZLÓ

Külső konzulens: -

Alk.	Dátum	Tartalom	Aláírás
1.	2023.10.02.	Fejlesztés tervezett menetének revíziója, szakmai egyeztetés, fejlesztési teendők megkezdése.	
2.	2023.10.30.	Fejlesztés állapotának és haladásának áttekintése. Technikai problémák és módosítások egyeztetése.	
3.	2023.11.20.	Felmerülő problémák egyeztetése és megoldása. Tesztelési fázis megindítása.	
4.	2023.11.30.	Dokumentáció véglegesítése. Leadással kapcsolatos teendők egyeztetése.	

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat II. tantárgy követelményét teljesítette, védésre bocsátható.

A konzulens által javasolt érdemjegy: 5

Sipos

....

Intézményi konzulens

Budapest, 2023.12.01.

TABLE OF CONTENTS

ABSZTRAKT

A kutatás célja egy potenciálisan innovatív intelligens forgalomirányítási rendszer feltárása, amely a forgalmat irányító rendőrök viselkedését utánzó intelligens közlekedési lámpák segítségével javíthatja a forgalomáramlást a kereszteződésekben. A rendszer vezérlési logikájaként egy "Right-To-Control" (RTC) mechanizmust javasoltam és validáltam, amely egy váltóverseny logikát alkalmaz, és egy adott időtartamra kijelöli az egyik közlekedési lámpát az egész csomópont "karmestereként", valós idejű bemeneti adatok alapján. A rendszer fő célja a forgalmi változásokra való valós idejű reagálás, ezzel kiküszöbölve a szükségtelen várakozási időt a zöld lámpáknál, ahol nincs ellenforgalom. A koncepció bizonyítására egy Unity-szimulációt fejlesztettem ki, amelyben különböző képzeletbeli és valós csomópontok kerültek lemodellezésre. Bár a rendszer kizárólag szimulált bemenetekre támaszkodik, a lehetséges gyakorlati alkalmazások kiemelése érdekében a valós érzékelők átfogó vizsgálata is bemutatásra kerül. Tesztelési céllal a rendszer működését különböző szimulált forgalmi helyzetek alapján értékeltem. Ezen átfogó értékelések optimális eredményeket mutattak, megerősítve a rendszer hatékonyságát a kitűzött célok elérésében. Mindazonáltal a rendszer funkcionalitásának fokozása érdekében további fejlesztési lehetőségek is vannak.

ABSTRACT

This research aims to unravel a potentially innovative smart traffic control system that can improve traffic flow at intersections using smart traffic lights that mimic the behavior of traffic controlling police officers. As the system’s control logic a "Right-To-Control" (RTC) mechanism was proposed and validated, employing a relay race logic that designates one traffic light as a "conductor" for the entire node for a specific duration determined from real-time input data. The ultimate goal of the system is to eliminate the unnecessary wait times at green lights with no opposing traffic by reacting to traffic changes in real time. As a proof of concept a Unity simulation was developed where different imaginary and real life intersections were modeled. While the system relies solely on simulated inputs, a comprehensive exploration of real-world sensors is presented to underline potential practical applications. For testing purposes the system’s operation was evaluated against different simulated traffic situations. The outcomes of these comprehensive evaluations revealed optimal results, affirming the system's efficacy in achieving its intended objectives. Nevertheless, there are potential avenues for further development to enhance the functionality of the system.

1. INTRODUCTION

Managing and reducing traffic congestion is one of the main issues of urban management. Traffic conditions are getting worse as a result of an increase in vehicle traffic in urban areas. Owing to this expansion, intersections get blocked and congested more often, costing commuters and goods deliveries time, particularly during rush hour. In addition, traffic congestion has a negative impact on the state's economy, ecology, and health, greatly impacting the quality of life especially in larger cities.
This detrimental effect can be mitigated with the use of smart systems that are able to react to the changing traffic conditions and direct the flow of traffic to reduce the number of vehicles idling at intersections.

As a frequent driver myself, I know all too well the feeling of impatience that builds up when waiting at an intersection with no opposing traffic in sight, wondering why the light won't turn green. This kind of scenario can cause not only frustration and wasted time, but also increased fuel consumption and CO2 emissions, contributing to the already pressing issue of climate change.
Being an environmentally conscious university student I feel it to be my responsibility to address these issues and given my field of study in computer science engineering I consider myself to be educated enough to investigate the solution of these problems.
While there are certainly numerous smart traffic control solutions in place today, I dare to explore a solution of my own by combining existing technologies and methodologies with my own ideas to make an optimized solution to replace traditional timed traffic lights.

I theorize that timed/scripted traffic systems are not equipped to meet present and future challenges of urban traffic controlling and they must be superseded by smart-node-based solutions mimicking the behavior of a traffic directing police officer while at the same time outsourcing some of the traffic flow related decisions to the drivers themselves.

The inspiration for my thesis project stems from my frequent experiences at a particular intersection (more on that later) during my commute home from the university, where I consistently observed the left turning lane becoming congested despite the absence of opposing traffic. This is due to the limitations of the traditional scripted traffic lights. I often found myself wondering how easy it would be if the light would just “know” that no opposing traffic is present and turning green could allow many people to be on their way.
Instead the light would have to wait for the pre-programmed timer to allow it to turn green more often than not resulting in the unfavorable situation where opposing traffic would arrive just before the turning lane received the green light.

Witnessing this almost everyday made me think about the solution and led to the inspiration of my thesis which is to experiment with the idea of smart traffic control signals and ultimately solve the above described problem through a Unity simulation.

In my study, I consider that allowing the individual driver to make a right turn on a red light is a safe and progressive way to simplify the decision making process resulting in less interruptions in the flow of traffic through urban areas.
Even when considering the pedestrian safety aspect [5] of this proposal I am confident that allowing drivers to make a right turn on red light (further on: Right on Red - RoR) would help mitigate traffic congestions and ultimately lead to a safer urban traffic environment.
According to the above considerations I will assume RoR as a legitimate driving maneuver in my modeling environment.

The scope of this thesis includes:

- Creating a 3D simulation in Unity
- Developing pseudo-artificial intelligence algorithm in C# to visualize and optimize the flow of traffic at road intersections
- Developing a modular node-based solution that allows a more complex network to be realized
- Proposing appropriate sensor array for providing real time traffic data
- Programming a simplified model of driver behavior
- Testing of different traffic scenarios to evaluate the effectiveness of the algorithms

2. LITERATURE RESEARCH

2.1. EXAMINATION AND EVALUATION OF SIMILAR SYSTEMS

2.1.1. An in-depth analysis of the Los Angeles Automated Traffic Surveillance and Control (ATSAC) System

The Los Angeles Automated Traffic Surveillance and Control (ATSAC) System[17] was an advanced, **hierarchical** traffic management system[Figure 2.1] with a supervisory computer at the Control Center and minicomputers at each ATSAC area controlling up to 400 intersections and 1600 detectors.

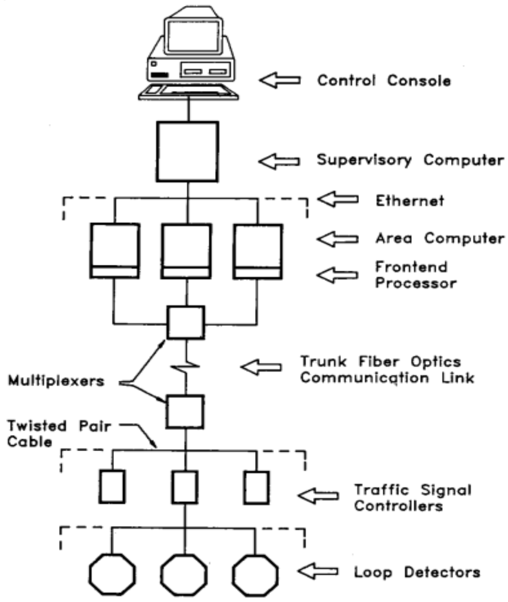


Figure 2.1 - The ATSAC System Architecture

The first phase of the ATSAC System was installed in Los Angeles, in June 1984. Front-end peripheral processing units (PPUs) and associated software had been developed to process control commands, surveillance data, and equipment status data. The system used a network of sensors, cameras, and computers to monitor and control traffic flow at intersections in real-time, with the goal of reducing congestion, improving safety, and enhancing mobility for drivers.

At that time the ATSAC system was considered to be a pioneering example of an intelligent transportation system (ITS), and was widely studied and emulated by other cities around the world. It was shown to be effective in reducing travel times, improving safety, and reducing emissions by reducing traffic congestion and smoothing traffic flow. The system continued to be upgraded and expanded over the years, and was then considered to be one of the most advanced traffic management systems in the world.

A variety of sensors and data sources had been used, including **loop detectors** embedded in the road, **video cameras**, and **GPS data** from connected vehicles, to collect information on traffic volume, speed, and congestion. This data had been analyzed by a central computer system, which had used algorithms and optimization techniques to adjust traffic signal timings and control the flow of traffic through the city's streets.

Loop detectors were placed in each marked approach lane at intersections of arterial streets, but at signalized intersections of an arterial street with a local street, semi-actuated signal control was used and there were usually no detectors on the arterial street at these locations. System detectors could be placed either in each marked lane at least 250 ft upstream of the signalized intersection or 100 ft on the departure side of the nearest upstream intersection which was signalized. The second option was more economical.
Real-time traffic surveillance data from the loop detectors was processed and transmitted for display on color graphic monitors at the Control Center. These monitors showed various static map displays generated by a computer, and the variable real-time information was passed from the supervisor computer to the graphics workstations. The displays on the color monitors showed the status of the control system equipment by depicting controller and detector status, and they also showed six traffic flow measures that were color-coded according to their magnitude.
Closed circuit television (CCTV) surveillance equipment was installed at critical locations to supplement the information from the electronic traffic detectors. The CCTV equipment was full motion color and could be remotely controlled by command from the ATSAC Control Center.

Communication techniques used:
- Trunk communications
- Fiber optics cable
- Multimode 6 fiber cable
- Single mode 24 fiber trunk cable
- Laser transmission at T1 data rate of 1.54 Mb/s
- Transmission over a city-owned 1200-Bd twisted pair telephone cable network
- Traffic flow and equipment status data transmission from each intersection to the communications hub using the same network

Four modes of control were used:
- time-of-day
- critical intersection control (CIC)

- traffic responsive
- manual override

Time-of-Day

Time-of-day plans were developed offline using traffic data and were fine-tuned by the ATSAC Control Center staff. Signal timing plans were created for different times of the day using the TRANSYT VII model and manual traffic counts. The ATSAC Control Center staff used color monitor displays of signal network operation and traffic surveillance data to fine-tune signal offsets and splits for newly developed time-of-day plans.

Critical Intersection Control

Critical Intersection Control (CIC) is an algorithm that adjusted the green time split at intersections in real-time based on traffic demand. Detectorization was required on each approach to CIC intersections to update the traffic demand equation every cycle. The green time was then prorated to each street based on relative demand, with limitations due to pedestrian clearance times. CIC maintained signal offset on only one street, which resulted in some performance degradation on secondary streets with closely-spaced signalized intersections.

Traffic Responsive Control

Traffic Responsive Control was an automated function that used a computer algorithm to select timing plans based on surveillance data. It was found to be more effective than time-of-day timing plans when day-to-day traffic variations were significant. The program also included a feature that prevented excessive timing plan switching by locking in a new plan for a minimum period of time after it had been selected. Additionally, the program limited plan selection during specific time periods or days to avoid extreme errors in plan selection.

Manual Override

The color monitor displays of traffic flow data were used to identify situations that required temporary manual override of the automated timing plans and to provide other information in the past. All traffic signals under computer control were directly addressable from the ATSAC Control Center, so the manual override could be applied to a single traffic signal, a group of traffic signals along a route, or whole areas of varying sizes. The manual override procedure required Control Center personnel to sort out from the large amount of information displayed those traffic situations that were beyond the bounds of automated control and then to quickly determine an appropriate manual override response.

A study was carried out to measure the enhancements accomplished through the use of the ATSAC System in the vicinity of the Coliseum in Los Angeles.

The results were as follows:
- stops reduced by **35%**
- intersection delay reduced by **20%**
- overall travel time reduced by **13 %**
- fuel consumption reduced by **12.5%**
- air emissions reduced by **10%**

Conclusion

At the time the system was effective, considering the available technologies. Using various sensors a semi-automatic traffic control system was made possible. However, today's systems are more sophisticated with state of the art sensors, and most of them are almost fully automated thanks to enhanced AI algorithms, making human intervention redundant. The ATSAC system still required humans to occasionally intervene with the operation of the system (Time-of-Day, Manual Override) which in my opinion introduces the possibility of human error while also slowing down the operation of the system.

2.1.2. Siemens Mobility

Siemens Mobility offers a range of intelligent traffic solutions, including adaptive signal control systems.
Their **Sitraffic Concert** product uses real-time data to adjust signal timing and optimize traffic flow. [18]
- **Real-time data processing**: The system uses real-time data from various sources, such as traffic sensors and cameras, to analyze traffic patterns and adjust signal timings accordingly.
- Dynamic signal control: The **system can adapt signal timings in real-time** to respond to changing traffic conditions, helping to reduce congestion and improve traffic flow.
- **Integrated traffic management**: The system can integrate with other traffic management tools, such as public transport information systems and emergency response systems, to provide a comprehensive traffic management solution.
- **Multi-modal** transportation management: The system **can manage** various modes of transportation, including **private vehicles, public transit, and bicycles**, to provide a more efficient and sustainable transportation network.

- **Predictive** analytics: The system uses predictive analytics to anticipate traffic patterns and adjust signal timings proactively.
- **Centralized management**: The system provides a centralized management interface that allows traffic operators to monitor and control traffic in real-time.

2.1.3. Econolite Centracs

Econolite Centracs[3] provides adaptive signal control systems for cities and municipalities. Their Centracs system uses **machine learning** algorithms to optimize traffic signal timing **based on real-time traffic data** provided by a wide variety of sensors:

Econolite CAvita is an intelligent transportation system (ITS) offered by Econolite that provides real-time traffic management and data collection capabilities. Some of the key features of CAvita include:

- Dynamic signal control: The system can adjust signal timings in real-time to respond to changing traffic conditions, reducing congestion and improving traffic flow.
- Integrated traffic management: **CAvita** can integrate with other traffic management tools, such as public transit information systems and **emergency response systems**
- Data collection and analysis: The system collects and analyzes real-time traffic data, including **vehicle counts, speed, and travel times**, to provide insights into traffic patterns and identify areas for improvement.
- Multi-modal transportation management: The system can manage various modes of transportation, **including private vehicles, public transit, and bicycles**, to provide a more efficient and sustainable transportation network.
- **Centralized management**: The system provides a centralized management interface that allows traffic operators to monitor and control traffic in real-time
- **Interoperability**: CAvita is designed to be compatible with a wide range of traffic management systems, making it easy to integrate with existing infrastructure and tools.
- **Support for connected and autonomous vehicles**: The system is designed to support connected and autonomous vehicle technology, enabling smoother and safer traffic flow.

Cloud-based Centracs® Mobility software platform now **supports crowdsourced traffic data** as an additional source of real-time roadway detection.[4]
The system allows traffic operators to incorporate real-time traffic data collected from **various sources, including connected vehicles, smartphones, and social media**, into their traffic management strategies.

2.1.4. Cubic Transportation Systems

Cubic Transportation Systems: Cubic offers a range of intelligent traffic solutions, including adaptive signal control.

SynchroGreen

The SynchroGreen[20] adaptive signal control system offered by Cubic Transportation Systems is designed to improve traffic flow and reduce congestion. The system uses real-time data and predictive analytics to adjust signal timings and optimize traffic flow.
Key features:

- Real-time **adaptive signal control**: The system uses real-time traffic data to adjust signal timings and optimize traffic flow in real-time.
- **Centralized management**: Synchro provides a centralized management interface that allows traffic operators to monitor and control traffic from a central location.
- **Data collection** and analysis: The system collects and analyzes **real-time traffic data**, including vehicle counts, speed, and travel times, to provide insights into traffic patterns and identify areas for improvement.
- **Multi-modal** transportation management: The system can manage various modes of transportation, including private vehicles, public transit, and bicycles, to provide a more efficient and sustainable transportation network.
- **Integration with other traffic management tools**: Synchro can integrate with other traffic management tools, such as public transit information systems and emergency response systems, to provide a comprehensive traffic management solution.
- Support for **connected and autonomous vehicles:** The system is designed to support connected and autonomous vehicle technology, enabling smoother and safer traffic flow.
- Performance metrics based on actual project results:

  40% Reduction in Arterial Travel Time
  35% Decreased Travel Time Average on a Typical Weekday
  38% Decreased Peak Hour Travel Time
  10% Reduction in Total System Delay, Saving Commuters Over **16,000 Hours** of Delay Annually

NextTraffic

The NextTraffic product uses **machine learning algorithms** to optimize signal timing and reduce congestion.

- Adaptive Signal Control: NextTraffic uses adaptive signal control to adjust signal timings in real-time based on current traffic conditions, reducing delays and improving overall traffic flow.
- **Intersection Monitoring**: The system monitors intersections using various sensors and cameras to detect the presence of vehicles and pedestrians.
- **Data Collection and Analysis**: NextTraffic collects and analyzes real-time traffic data, including vehicle counts, speed, and travel times.
- **Multi-modal Transportation Management**: The system can manage various modes of transportation, including private vehicles, public transit, and bicycles.
- **Connected and Autonomous Vehicle Integration**: NextTraffic is designed to support connected and autonomous vehicle technology.
- **Cloud-Based Central Management**: The system provides a centralized management interface that allows traffic operators to monitor and control traffic from a central location, with cloud-based hosting options to reduce hardware and maintenance costs.

2.1.5. Comparison of features and services of ITS providers

Ranking method explained:
I decided to use a weighted ranking method[6] as a decision-making technique involving assigning weights or importance values to different criteria or factors, and then using those weights to score and rank the various providers. I identified the criteria, assigned weights to each of the criteria based on their relative importance in the decision-making process
Weights are expressed as:

1. less important or considered legacy technology/method
2. important, up-to-date technology/method
3. future-proof, innovative technology/method

Scores are based on the cumulative weights of each provider.
Finally, the subjective ranking ranks providers in order of their total score, with the highest ranking provider being the preferred technology provider.
By using the weighted ranking method, I can ensure that all relevant factors are considered and weighing them appropriately helps an informed and effective decision-making.
Empty table cells represent data that is not available or ambiguous.

Note: All ITS Providers examined in this study use adaptive signal control. For that reason the comparison is omitted.

Provider		Siemens Mobility	Econolite	Cubic Transportation Systems	
System brand name		Sitraffic Concert	CAvita	SynchroGreen	NextTraffic
Collects real-time traffic data via	W*				
Radar technology[1]	2	yes	EVO RADAR™	yes	yes
Above-ground cameras[2]	2	yes	AUTOSCOPE® VISION	yes	yes
In-ground magnetometers/ Inductive loop detection[3]	1	yes	ACCUSENSE®	yes	yes
GPS data from vehicles[4]	3	yes	yes		
Bluetooth beacons	3	no	no	no	no
Uses data from public transportation schedules	2			yes	
Uses data from emergency response systems	2	yes	yes		
Uses crowdsourced/social media generated/web scraped traffic data	3	no	yes		
Multi-modal[7]	2	yes	yes	yes	yes
Pedestrian detection	2				yes
Incident detection/management	2		yes		yes
Supports connected and autonomous vehicle integration	3	no	no	yes	yes
Supports integration with other ITS	2	yes	yes	yes	yes
Central management architecture (on-premises)	1	yes	-	yes	-
Central management architecture (cloud-based)	3	-	yes	-	yes

Uses machine learning (ML)	3	no	yes	yes	yes
Uses predictive algorithms	3	yes			yes
Uses V2I technology[8]	3	no	no	no	no
Evaluation		One of the key benefits of Sitraffic Concert is its ability to provide real-time information to drivers, enabling them to make informed decisions about their route and avoid congested areas. The system can also be used by traffic management centers to monitor traffic conditions and adjust traffic signals and signs to improve traffic flow and safety. Sitraffic Concert is a powerful and effective tool for managing traffic in urban areas. One of the most widely used ITS. The lack of support for crowdsourced data might negatively affect its effectiveness in the future.	Centracs integrates a variety of data sources including sensors, cameras, and GPS data from vehicles to provide real-time information about traffic conditions. The system includes features such as incident detection and response, and performance monitoring and reporting. Centracs also includes tools for incident management, allowing transportation agencies to quickly detect and respond to incidents such as accidents or road closures. Centracs is a powerful and effective transportation management system. The system's ability to integrate multiple data sources and provide real-time information makes it a valuable tool for transportation agencies.	SynchroGreen is designed to work with existing traffic signal infrastructure, using a combination of sensors and data sources. SynchroGreen can adjust signal timing based on traffic flow, pedestrian activity, and public transportation schedules. Can be configured to prioritize emergency vehicles or public transportation systems. SynchroGreen also includes tools for monitoring and reporting on traffic performance, allowing transportation agencies to identify problem areas and make data-driven decisions about future infrastructure improvements.	NextTraffic uses a combination of real-time traffic data and predictive analytics to monitor traffic conditions and make adjustments to traffic signals and other transportation infrastructure. The system can also provide real-time traffic information to drivers and public transportation systems. NextTraffic also includes tools for incident management, allowing transportation agencies to quickly detect and respond to incidents such as accidents or road closures. NextTraffic is a powerful and comprehensive transportation management system.
Score		18	25	19	25
Subjective ranking		4	2	3	1

Table 2.1 - Comparison and ranking of ITS solutions

[*] Weight of the feature when considering the overall subjective ranking.

[1] Devices transmitting and receiving radar signals to calculate the range, velocity, and direction of targets.

[2] Capture real-time footage of traffic conditions, allowing the system to monitor traffic flow, detect incidents, and analyze traffic patterns.

[3] Devices embedded in the roadway and detect the presence of vehicles by measuring changes in the magnetic field.
[4] GPS data: Can be collected from vehicles and used to track traffic speed and movement, and to provide real-time traffic information to drivers.
[5] Pedestrian sensors: Detect the presence of pedestrians at crosswalks and can be used to adjust signal timing to improve safety and mobility for pedestrians.
[6] Provides real-time traffic information directly and individually to road users on their mobile devices.
[7] Able to manage private vehicles, public transit, and bicycles
[8] Enable vehicles to communicate with transportation infrastructure (V2I)

Conclusion

According to this research I found Cubic Transportation Systems' NextTraffic system being the closest to the requirements of being a modern future proof traffic control solution. However, my approach will use few existing solutions from either of the systems detailed above.

2.1.6. Unity and Unreal Engine

Unity

Since my system is being developed in Unity[25], I find it important to present here the features of Unity and explain why I decided to choose Unity against Unreal Engine.

Unity is a popular cross-platform game engine that has gained widespread use in game development and other industries.[27] It provides developers with a powerful set of tools and features for creating immersive, interactive experiences that run on a wide range of devices and platforms. Unity offers a range of features for creating realistic and engaging simulations. It includes a physics engine that can simulate realistic physical interactions between objects, as well as a built-in AI system that can be used to create intelligent NPCs and other interactive elements.

Unity supports multiple programming languages including: C#, C++ and JavaScript, with C# being the default which makes it a great tool for the majority of developers, including beginners. It is also the preferred programming language for me as I am used to C# more than any other programming languages. Also C# is well-suited for developing simulations of smart traffic systems, as it allows the creation of complex algorithms and data structures that can be used to model the behavior of traffic and other elements of the simulation.

Unity also has an online marketplace called the Asset Store where developers can buy, sell, and share a wide range of assets, including 3D models, animations, scripts, and plugins. It offers a convenient way to quickly and easily find and integrate high-quality assets into projects, without having to create everything from scratch. This comes in handy for me, due to the size of my project. Not having to build everything from the ground up helps a great deal in developing each layer of my project.

A minor but important factor was the community behind the engines. Unity's large and active community makes it easier to find solutions to potential problems, resulting in a smoother development process.

When I was deciding what environment to use for developing the simulation another game engine came up: Unreal Engine 5[26].
Unreal Engine 5 (UE5) is the latest version of the popular game engine developed by Epic Games. It provides developers with a range of powerful tools and features for creating immersive, interactive experiences, including games, simulations, and other real-time applications.
While UE5 is undoubtedly a great tool for creating my thesis project in some ways it is an overkill compared to Unity. For example my simulation will not use such heavy graphics like high-quality lighting, shadows, and reflections that UE5 was meant for.
Ease of use was also a consideration of mine when deciding on the platform. Unity is generally considered easier to use and more beginner-friendly than Unreal Engine 5. Its visual interface and drag-and-drop functionality make it easy to create scenes and animations. In contrast, Unreal Engine 5 has a more complex interface.

In conclusion my choice falls on Unity as it has a much less steep learning curve and better suited solution for my project.

2.2. TRAFFIC CONTROL

2.2.1. Traffic police officers

Traffic control officers play a critical role in maintaining public safety during road disruptions, accidents, and public events. They ensure the orderly movement of vehicle and pedestrian traffic using various tools, including signs and signals. Their duties involve coordinating the movements of emergency response teams, construction workers, and the general public. They operate at construction zones, streets, highways, and other locations in urban, suburban, and rural areas. [16]

Traffic police have been in existence for almost three centuries, with some records suggesting that the first traffic police force was established in London in 1722. In the early days they used hand signals and whistles to direct the flow of traffic and to ensure public safety on the roads. Since then their methods evolved over the years but their fundamental purpose remained more or less the same, which is to direct traffic in intersections. This is usually done by a single officer using hand signals[Figure 2.2] and a whistle. Today we might see traffic officers use a portable traffic signal, often in the form of a stop sign, or a handheld light stick. [11]

Hand signals used by traffic controlling officers:

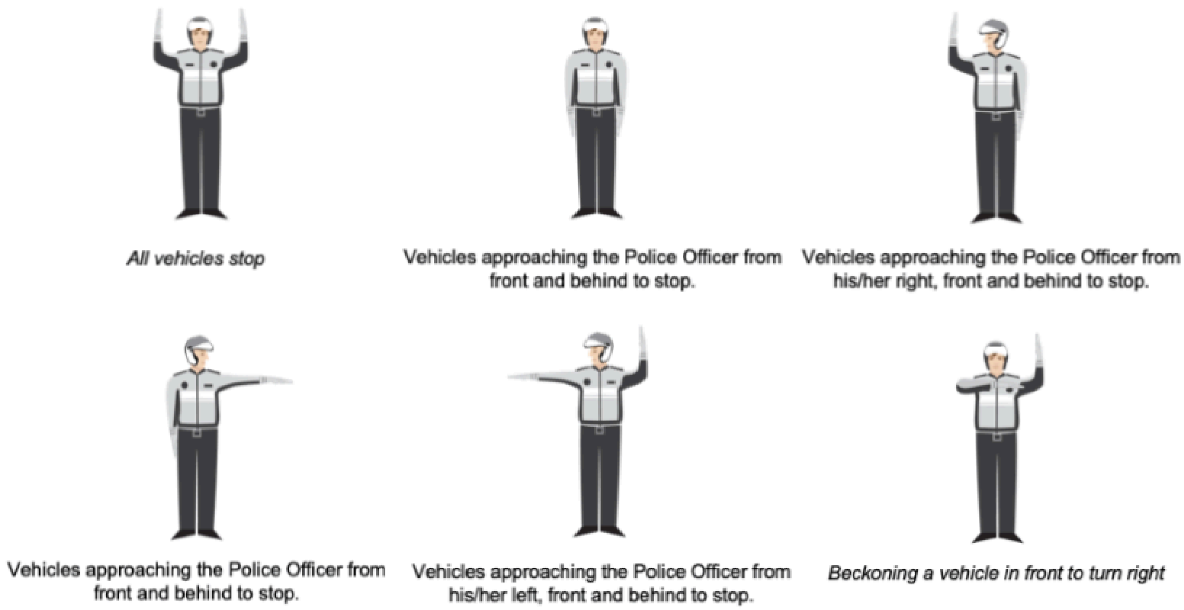


Figure 2.2 - Police Officer Hand Signals

The relevance of traffic controlling officers will be detailed later in my thesis, where I explore my idea of combining the fundamentals of their operation with traditional and smart traffic control systems.
While the decision making capabilities of the officers are superior to the traditional scripted traffic lights, they are still human, which introduces human error. Humans get tired over time, which affects their speed, precision and decision making capabilities. They are less capable of handling intense situations quickly and humans can process the immense data that comes from the environment much slower than a computer with a sophisticated algorithm. All of these lead to potential human errors which have a negative impact on traffic.

For that reason, my idea is to mimic the decision making capabilities of traffic officers and incorporate it into a smart traffic control system that will use a variety of tools, including sensors to gather data and smart algorithms to make decisions in real-time. By doing so, the potential for human error, which can result in accidents and traffic congestion, is eliminated.

2.2.2. Traffic lights

The history of traffic lights dates back to the late 19th century, when the first traffic signal was invented in London in 1868. This signal consisted of two gas lamps that could be manually illuminated by a police officer, indicating whether vehicles should stop or proceed with caution. However, the system was not perfect, as the lamps were often obscured by fog and could cause explosions if gas leaked. Over the next few decades, various inventors and engineers tried to improve upon the traffic light concept by using electricity, colors, and timers.[1][21]

In 1914, the first electric traffic signal was installed in Cleveland, Ohio. It featured red and green lights that alternated with a buzzer sound, and this design became the standard for many years. However, traffic lights continued to evolve with new features such as amber or yellow lights as a warning signal, pedestrian signals to protect walkers, flashing lights and arrows to indicate turning lanes and traffic flow, and more sophisticated circuitry and controls that could adapt to changing traffic patterns and reduce congestion.[1][21]

The meaning of the colored traffic lights has also evolved over time. Today, red indicates stop, yellow indicates caution or slow down, and green indicates go. However, in the early days of traffic lights, red meant stop, green meant caution, and white meant go. The switch to green for go was made to better align with the universal understanding of green as a positive signal.[22]

Today's traffic lights are highly advanced systems that use a combination of sensors, controllers, and communication technologies to manage traffic flow. The main components of a modern traffic light include the:

- signal head
- controller
- detector

The signal head is the most visible part of the system and typically consists of multiple lights arranged in a specific pattern.
The controller is the brain of the system and uses sensors to detect the presence of vehicles and pedestrians and adjust the signal timing accordingly.
The detector is the component that senses the presence of vehicles or pedestrians at an intersection.

Despite the increasing prevalence of smart traffic systems worldwide, most intersections still rely on pre-programmed timing signals for traffic lights.[12][24] The timing signals for traffic lights are set based on a variety of factors, including:
- the volume of traffic
- the presence of pedestrians
- the time of day

In general, the goal of timing the signals is to balance the competing demands of safety, efficiency, and mobility. For example, during rush hour, the signals may be timed to allow for longer green lights and shorter red lights to reduce congestion and improve mobility.[12][24] Conversely, during off-peak hours, the signals may be timed to prioritize safety, allowing for longer red lights and shorter green lights to ensure that pedestrians can cross the intersection safely.

The factors that influence the operation of traffic lights are numerous and varied. In addition to traffic volume and pedestrian activity, other factors that can affect the operation of traffic lights include:
- weather conditions
- emergency vehicles
- special events

For example, during inclement weather, the signals may be adjusted to allow for slower traffic and increased stopping distances. Similarly, during special events such as parades or festivals, the signals may be timed to accommodate higher pedestrian volumes and increased traffic flow.

While these methods of signal timing have been in use for decades and proved to be effective, they are only effective to some degree. Hard coded times for signal change have both positive and negative effects on traffic control.

A positive effect could be that due to the fixed timing of color change (red light, green light) a particular intersection would be predictable to frequent drivers making them less inclined to run a red light, since they know how much they have to wait for the next green light. Conversely the opposite could be true, if the intersection has a much longer red light than green light, a driver might try to skip the long wait by running a red light.

Red light running[14] is defined as entering an intersection after the signal light has turned red. This is considered a violation, although drivers who are already in the intersection when the signal changes, such as those waiting to turn left, are not red light runners. In some areas, failing to come to a complete stop before turning right on red, or turning right on red where it is prohibited may also be considered red light running.

Red light running is one of the leading causes of accidents on the road that lead to fatalities and severe injuries. "According to data from the National Safety Council, **42,060** lives were lost on U.S. roads in **2020**. Between **2008 and 2019**, the **National Highway Traffic Safety Administration** estimated **9,227** people were killed in crashes related to red-light running. **In 2019** alone, more than **840** people died due to red-light running." [15]

A negative effect of hard coded timing signals is that the traffic control system cannot handle traffic dynamically meaning that even if no opposing traffic is present, vehicles still have to wait for the light to turn green and allow the flow of traffic. The unnecessary wait is not only unpleasant for the drivers, but raises a major consideration for the environment. The emitted CO2 of vehicles contributes a large amount to the worsening situation of climate change. In fact, transportation is the leading CO2 emission sector in the world[Figure 2.3], according to the United States Environmental Protection Agency (EPA).[19]

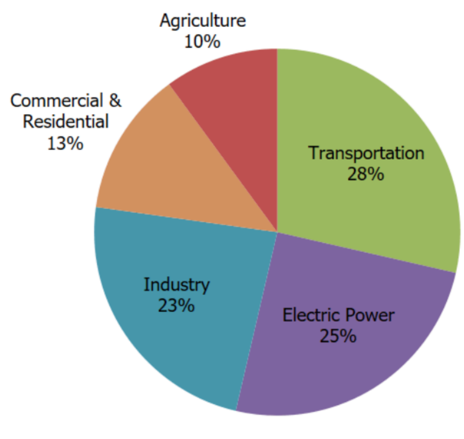


Figure 2.3 - CO2 Emission by Sectors

On average, a typical passenger vehicle emits approximately **4.6 metric tons** (4,600 kilograms) of carbon dioxide per year. This is based on the assumption that the average gasoline vehicle on the road today has a fuel economy of about **9.4 liters per 100 kilometers** and drives around **18,500 kilometers per year**. It is important to note that every liter of gasoline burned generates about **2.3 kilograms of CO_2**. [7]

2.2.3. Vehicle detection methods used in modern traffic control systems

While my system is solely a simulation and does not directly incorporate real-world sensors, I found it important to outline some currently utilized sensors. Given that my system's logic (further discussed in Chapter 5.2.3), solely relies on inputs from sensors, any of the following sensors could be employed. This establishes an abstracted layered approach.

Inductive Loop Detection

Inductive loop detectors[13] work by utilizing a loop of wire that is embedded in the road surface to detect vehicles passing over it.[Figure 2.4] When a vehicle passes over the loop, the metal components of the vehicle disrupt the electromagnetic field created by the loop, causing a change in the inductance of the wire. This change is detected by an electronic circuit connected to the loop, which then sends a signal to a traffic controller or other monitoring device indicating that a vehicle has passed over the loop.

The sensitivity and accuracy of the inductive loop detector depend on the size and shape of the loop, the frequency of the electromagnetic field, and the type of metal used in the wire. The detection zone of the loop can be adjusted by changing the number of turns in the wire and the frequency of the electromagnetic field.
Inductive loop detectors are commonly used in smart traffic control systems to monitor traffic flow, control traffic lights, and gather data for traffic analysis. They are relatively low-cost and reliable, but they require periodic maintenance to ensure proper functioning.

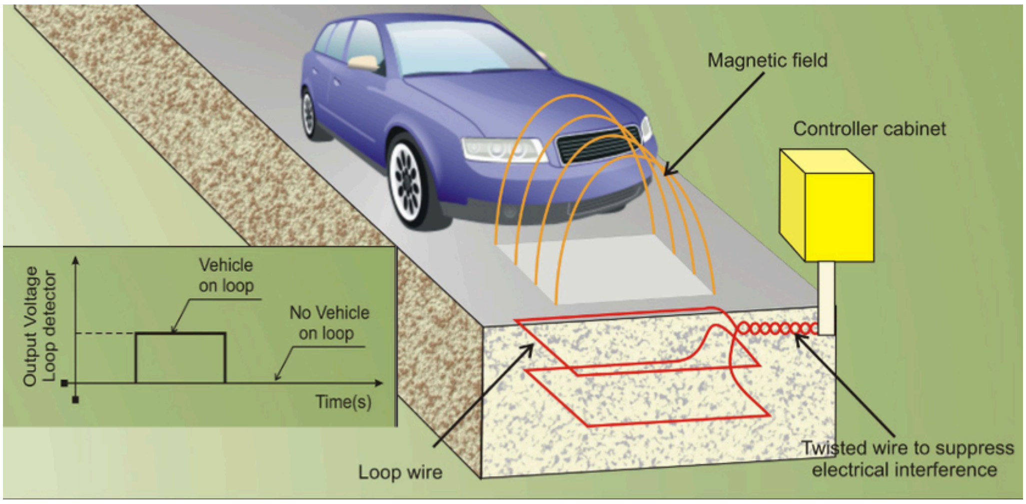


Figure 2.4 - Inductive Loop Detection

Pros:

- high accuracy in detecting vehicles
- can distinguish between different types of vehicles (motorcycles, cars, trucks, etc.)
- capable of detecting vehicles in almost any weather condition and at all times of the day
- can be integrated with other traffic management systems to improve overall traffic flow

Cons:

- expensive to install and require significant infrastructure (pavement cut, lane closure)
- can be affected by environmental factors such as
  - temperature changes
  - moisture
  - electromagnetic interference
- limited coverage and may not detect vehicles in certain areas
- prone to damage from pavement wear, roadworks

Infrared Detectors

Infrared sensors[2] work by detecting the heat signature, or infrared energy, emitted by objects. These sensors are commonly used in traffic control systems to detect the presence of vehicles and trigger traffic signals accordingly.
Infrared radiation is a type of electromagnetic radiation that has longer wavelengths than visible light. It is emitted by any object that has a temperature greater than absolute zero and is used in a variety of applications such as remote temperature sensing, communication, and sensing of motion.
The range of infrared radiation is between 700 nanometers to 1 millimeter in wavelength, which corresponds to frequencies from 430 THz to 300 GHz, respectively.[Figure 2.5]

However, the practical range of infrared sensors depends on factors such as the type of sensor, the atmospheric conditions, and the sensitivity of the sensor. In general, the range of infrared sensors used in traffic applications is usually within a few meters to several tens of meters. [10]

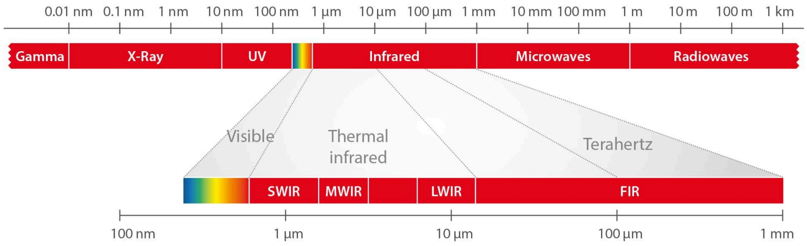


Figure 2.5 - Light Spectrum Interval

There are two types of infrared sensors used in traffic control systems: **passive** and **active**.

A passive infrared (PIR) sensor[2][Figure 2.6] detects infrared radiation emitted by objects in its field of view without emitting its own infrared light. It works by detecting the heat signatures of objects in the form of infrared radiation. They are considered passive because they do not emit any energy, but instead, they detect and respond to energy emitted by other sources, such as human bodies or vehicles.

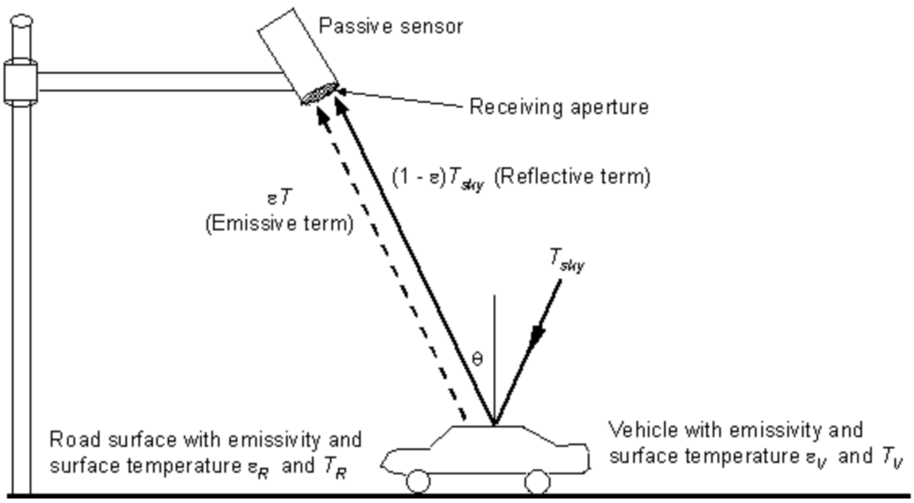


Figure 2.6 - Passive Infrared Sensor

An active infrared (IR) sensor[2][Figure 2.7] is a type of IR sensor that emits IR radiation to detect the presence of objects or vehicles. It emits a beam of IR radiation and then measures the amount of radiation that is reflected back to the sensor. If an object or vehicle is present, the amount of reflected IR radiation will be different than if there is no object or vehicle present.

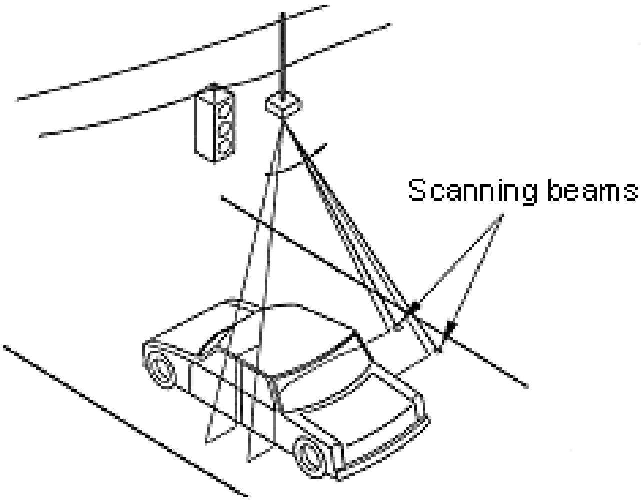


Figure 2.7 - Active Infrared Sensor

Infrared detectors[2][8] offer several benefits such as detecting temperature changes and motion, and being less influenced by visible light. They are also cost-effective and versatile, with applications in security systems, temperature sensors, and motion sensors.

Despite their advantages, infrared detectors have some drawbacks.[2] One limitation is their restricted range, as they can only detect objects within a specific distance. In addition, environmental factors such as humidity, ambient temperature, and other conditions can affect their accuracy. Also, some materials can interfere with the detection of infrared radiation by absorbing or reflecting it.

Video Camera Systems

Video camera systems detect and control traffic by analyzing live video feeds of the road.[Figure 2.8] The cameras capture high-resolution images of the road and surrounding areas, and these images are analyzed in real-time using advanced computer vision algorithms.[9]

The computer vision algorithms can detect vehicles, pedestrians, and other objects in the video feed and track their movements. They can also detect traffic flow, vehicle speed, and congestion. This information is then used to adjust traffic signals in real-time, to optimize traffic flow and reduce congestion.[9]

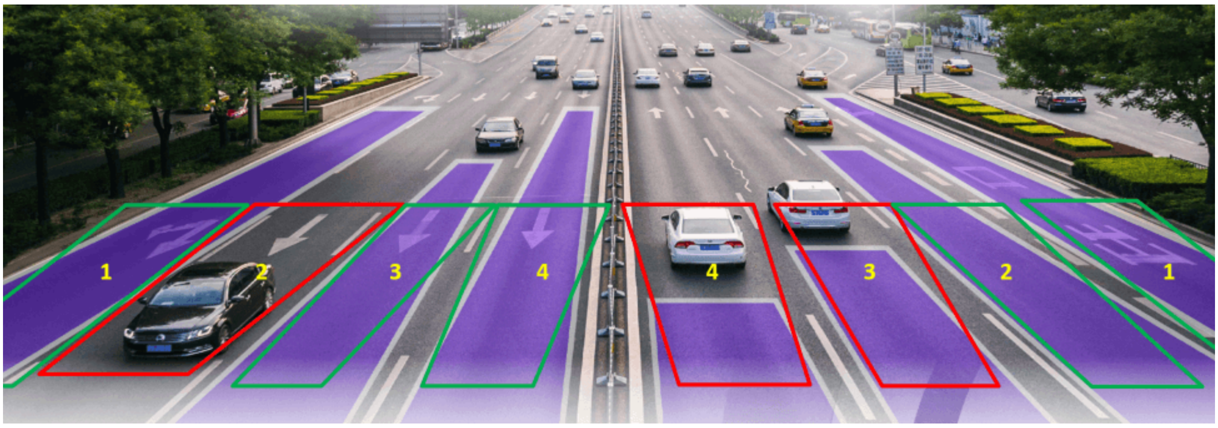


Figure 2.8 - Vehicle Detection via Video Camera

These systems are often connected to data centers that can handle the processing and storage of large amounts of data. Data centers receive data from the video camera systems and use advanced analytics tools to process the data and generate insights. These insights are then used to adjust traffic signals in real-time, to optimize traffic flow and reduce congestion. Additionally, data centers can also store the video footage from the camera systems for future analysis, which can be used to identify areas of improvement and make adjustments to the traffic control system. The data center is an integral part of the smart traffic control system, as it provides the computing power and storage needed to analyze and process the large amounts of data generated by the video camera systems.[23]

One of the main advantages of video camera systems is that they are highly accurate and can detect a wide range of objects and events in real-time. They can also be used in a variety of weather conditions and lighting conditions.[9]

However, there are also some limitations to video camera systems. They can be expensive to install and maintain, and they require significant computing power to analyze video feeds in real-time. Additionally, they can raise concerns about privacy and surveillance.

2.3. SUMMARY OF LITERATURE RESEARCH

Today's traffic control infrastructure has significantly evolved to incorporate a diverse range of elements that work together seamlessly to ensure smooth traffic flow and enhanced road safety. In addition to the traditional presence of traffic control police officers stationed at key intersections advanced technologies have been integrated to optimize traffic management. One vital component of modern traffic control infrastructure is the implementation of hard-coded and scripted traffic signals. These signals are strategically placed at intersections and operate based on predefined patterns or time intervals. By adhering to a predetermined sequence, these signals help regulate the flow of vehicles, minimizing congestion and ensuring efficient movement on the roads.

However, the reliance on fixed traffic signals alone is gradually shifting towards more intelligent and adaptive systems. Real-time data-based smart traffic systems have emerged as a game-changer in traffic control infrastructure. These systems leverage advanced sensors, cameras, and communication networks to gather real-time data on traffic conditions.

Moreover, the integration of smart traffic systems with emerging technologies like artificial intelligence (AI) and machine learning (ML) holds great promise. These technologies enable traffic control infrastructure to learn from historical data, predict traffic patterns, and optimize traffic management strategies proactively.

According to my research there are a number of viable smart traffic control solutions on the market. They are similar in the sense that they all use some kind of real time detection method (see Comparison and ranking of ITS solutions table) to dynamically control the traffic together with intelligent algorithms and prediction methods. They are either centralized/decentralized or hybrid systems with either brick and mortar or cloud based central control hub backends. Different manufacturers' systems can be connected to each other to allow interoperability.

Based on my literature research I concluded that the current systems, no matter how advanced or refined they are, all lack the paradigm changing approach that my proposal could inherit from.
Therefore, I decided that I will structure my solution around the following key features:

- Node-based (not centralized) decision making units - I will refer to these as intersection nodes
- Simulating the problem solving capacity of a single traffic control police officer per intersection node - decision making is done by each intersection individually and not through a common, centralized control unit
- Up to date detection methods such as infrared sensors, inductive loop detection and intelligent video surveillance boiling down to a simple data set consisting of two input fields:
    - how many vehicles are waiting at the individual traffic lights
    - for how long they have been waiting

In my thesis, I will delve into the advantages and disadvantages of my unique approach. I intend to explore these aspects through simulation and testing methodologies. By utilizing these methods, I aim to gain valuable insights into the viability and potential limitations of my approach.

3. SPECIFICATION REQUIREMENTS

Minimum waiting time for vehicles

The most important aspect of my system is that it should be able to ensure the minimum waiting time for vehicles passing through an intersection. To achieve this, it is essential for the interconnected smart lights within the node to collaborate harmoniously, effectively coordinating the flow of traffic and maintaining equitable waiting times for vehicles in each lane.
For this purpose a pseudo road priority ranking method will be used to establish a lane hierarchy identifying those that are of utmost importance to maintain minimal congestion compared to others. This will be important in the decision making of the lights. The actual algorithm will be detailed in the system design part of my thesis.

Decentralized decision-making

The whole system is built on the idea of mimicking the decision making capabilities of a traffic controlling officer, thus **segregating the individual nodes from each other.** This concept allows the nodes to only base their decisions on traffic conditions that are relevant to that specific node. Controlling traffic this way eliminates the need for a centralized computing unit to make decisions for all the nodes in a geographical area (e.g. a city).

More complex systems from the nodes

In my simulation one of the functions of a working node is that it should be able to connect to other nodes, thus creating a more complex system. Each node is responsible for only their vicinity traffic wise and connecting them should solve the optimisation of traffic flow in a larger area.
This modular approach simplifies the development, deployment and management of the infrastructure, because modifications will only have to be made to the base-node. Moreover, different intersections can be made from the same base model for testing purposes and simulation of real life examples.

4. PLANNING

4.1. SYSTEM DESIGN

4.1.1. System Components

- **Node**
  - the encapsulation of all parts required to operate one intersection
- **Intersection**
  - two or more roads crossing with alternating flow of traffic controlled by a traffic light
  - can be further dissected into discrete atomic intersections, each serving as a fundamental building block for the overall system.
- **Atomic Intersection**
  - either 'X' or 'Y' type crossing of roads
  - cannot be further simplified
- **Traffic Light**
  - controls flow of traffic by allowing or restricting vehicle movement on a given road / direction
  - green light, red light (yellow is omitted for simulation purposes)
- **Vehicles**
  - individual moving objects for traffic simulation
  - passive object - will not communicate with the traffic control system
- **Traffic Generator**
  - controller feeding the vehicles into the intersection
- **Road**
  - contains multiple lanes
- **Lane**
  - path of vehicles, may lead straight or turn left or right

Terminology

- **Road Priority - This will be deprecated during Development**
  - a single number identifying which lane takes precedence over the others
  - roman numbers: I, II, III (one being the highest priority)
- **Road Saturation**
  - number of vehicles allowed waiting in front of a traffic light at any given time
  - not to be confused with the actual vehicle capacity of a road
- **Maximum Waiting Time**

    - the maximum waiting time allowed for the first vehicle arriving at a traffic light based on priority (hard coded value in seconds)
- **Green Light Request**
    - event to request green light from traffic lights with restrictive relation
    - event triggers either when
        - road saturation reached max limit
        - maximum waiting time has been reached
- **Control Cycle**
    - one full sequence of lights: red → green → red
- **Restrictive/Permissive Relation**
    - permissive relation: two traffic lights are allowed to be green at the same time - used where one flow of traffic does not interrupt/ the other traffic flow
    - restrictive relation: when one of the lights is green the other must be red and vice versa
- **Traffic Flow Direction North/East/South/West**
    - as a convention cardinals are denoting the direction the traffic flows from e.g. North traffic arrives from the north

4.1.2. Atomic Nodes

Each intersection can be dissected into two types of distinct elements called atomic nodes for the purpose of this thesis: X [Figure 4.1] and Y [Figure 4.2] type nodes. With the combination of these atomic nodes all intersections can be built regardless of its complexity. The following figures give a schematic overview of their distinct features.

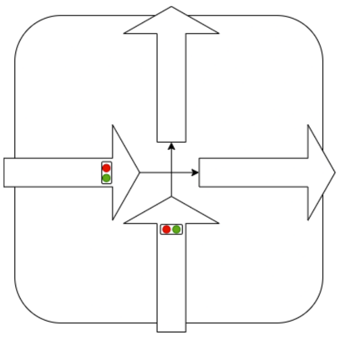


Figure 4.1 - Atomic X-Node

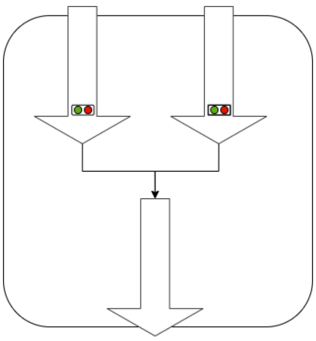


Figure 4.2 - Atomic Y-Node

Although an X-node topologically could be constructed using two Y nodes, I decided to use them as separate entities for they have different saturation properties.
While in an atomic X-node the saturation of an egress road changes according to the following rules: it either remains the same in the case of all the cars leaving the intersection on

the same route or decreases in case of cars leaving on separate routes, however the two ingress roads of a Y-node increases the saturation of the single egress road in all cases.
Example of combining an X-node and a Y-node[Figure 4.3] where the egress road of the Y-node feeds the ingress road of the X-node. In this case all other ingress roads are fed by Traffic Generators.

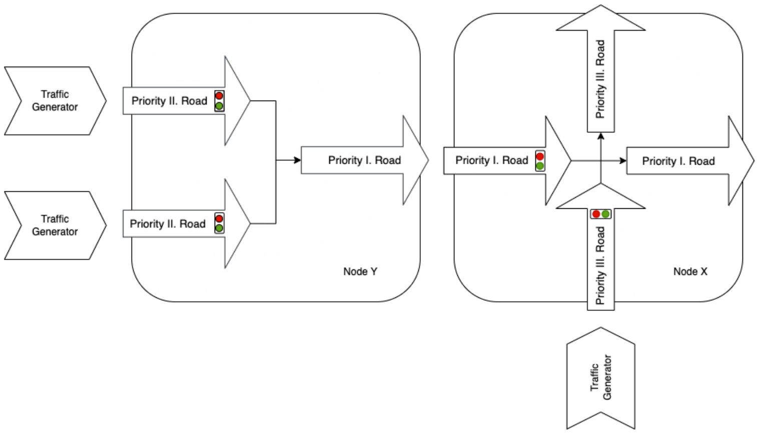


Figure 4.3 - System Schematics Combining X and Y Nodes

4.1.3. System Schematics

The following figure [Figure 4.4] is an overview of the relationship of the nodes and their elements.

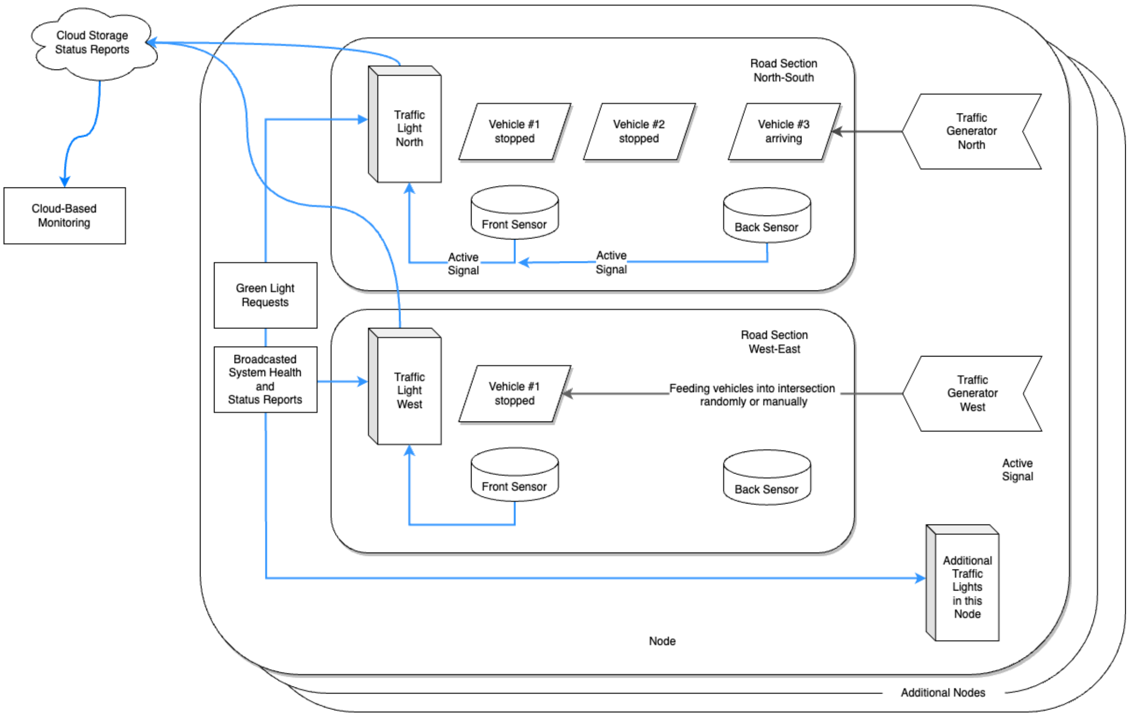


Figure 4.4 - System Schematics Block Diagram

4.1.4. Control Flow of an Individual Traffic Light

The following control flow diagram[Figure 4.5] represents the events of a new vehicle arriving in an intersection from the point of view of the traffic light.

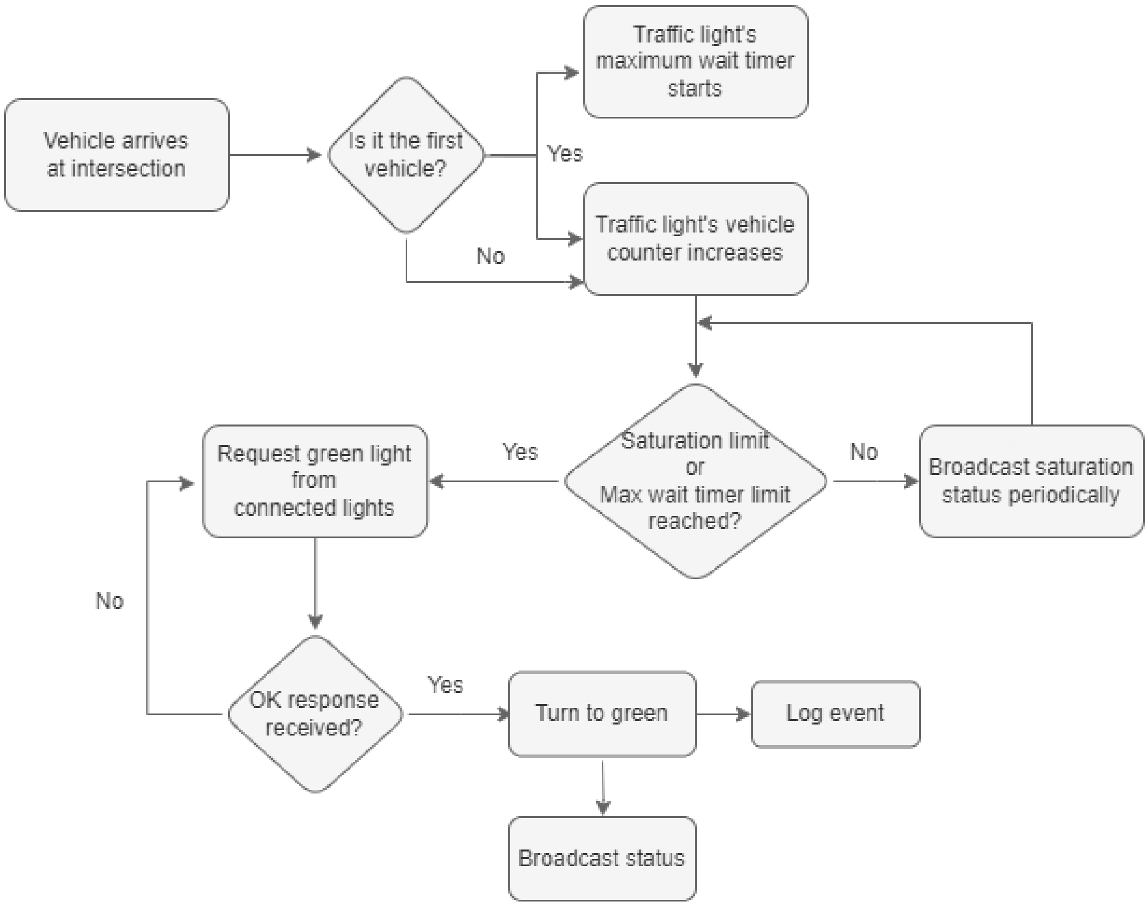


Figure 4.5 - Planned Control Flow Diagram of an Individual Traffic Light

4.1.5. Control Light Sequence

In the following table[Table 4.1] an example of a full control cycle can be seen.

	A light - I. priority	B light - III. priority
Control Cycle Starts		
0s	green	red
5s		broadcasts status 1 car arrived no max saturation reached
10s	ignores B light’s 1st green request based on road priority	+3 cars arrived max saturation reached requests green light
15s	ignores B light’s 2nd green request based on road priority	requests green light
20s	receives request from B light to turn green turns red responds OK which forces B light to turn green (restrictive relation)	requests green light receives OK from A light turns green
Half Way Through Control Cycle		
0s	red	green
5s	broadcasts status 10 cars arrived no max saturation reached	
10s	+10 cars arrived max saturation reached requests green light	receives request from A light to turn green turns red responds OK
	turns green	red
Full Control Cycle Ends		

Table 4.1 - Control Light Sequence

Control Light Sequence Diagram

The following diagram [Figure 4.6] outlines a full control cycle of two connected lights.

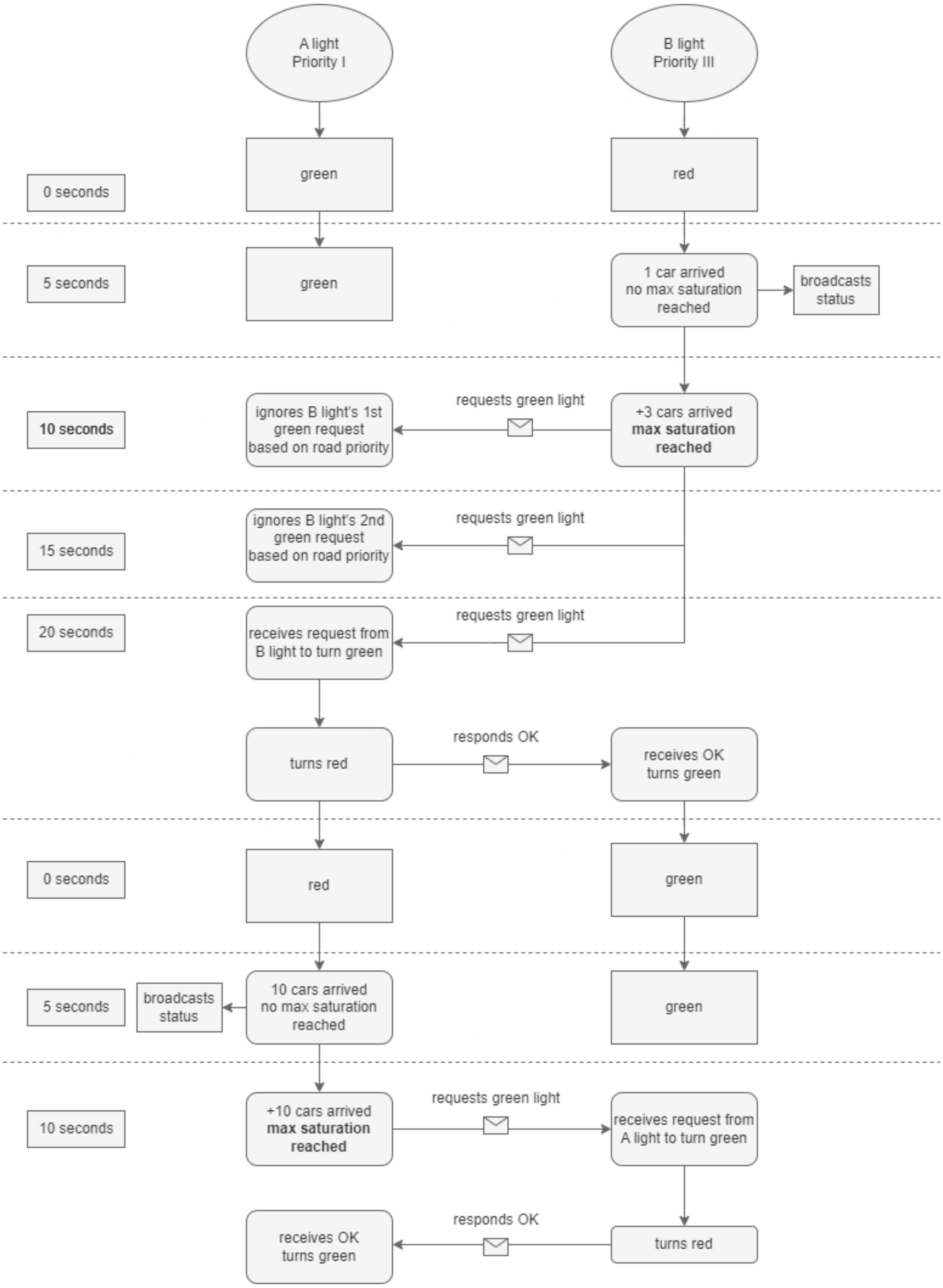


Figure 4.6 - Control Light Sequence Diagram

4.2. LIST OF FUNCTIONS

- Traffic lights equipped with decentralized decision-making capabilities:
  - the lights should be able to make decisions without a centralized computing hub in real time
- Interface to monitor the traffic lights:
  - each node element should be able to report its current operation status (red / green) and health status (light is functioning properly or out-of-order)
- Communication between the smart-traffic lights:
  - for proper functioning of the node, the individual traffic signals must be able to communicate with each other
  - request data packages include:
    - header:
      - identifier of individual lights
      - timestamp of request
    - health info: to make sure that the request is sanctioned and valid
    - Green Light Request level
      - timed
      - based on road saturation
- Working algorithm for the decision making:
  - pseudo artificial intelligent algorithm to decide which mode the light should be in (red / green) based on input data from the sensors
- Sensors providing input data for the algorithm:
  - feeding the required data to the traffic signal about its designated lane
  - data include: number of vehicles, types of vehicles, waiting time of the first vehicle
- Simulation of realistic traffic:
  - vehicles randomly fed into the system based on road priority
- Interconnection of individual nodes:
  - creating complex networks of intersections should be possible by combining the created base node with copies of itself
  - simulating existing intersections
- Consistent functioning in different traffic situations
  - the system should be able to perform optimal traffic control in different traffic situations proving its dynamic decision making capability
  - different situations may include a lower priority lane being more congested than higher priority lanes → the lights should recognize this and adjust themselves accordingly
- Logging: Traffic lights have to log their actions for security reasons
  - timestamps of state change
  - unique identifier

- after state change:
  - number of vehicles that were waiting
  - length of red light
  - length of green light

4.3. PLANNED COURSE OF DEVELOPMENT

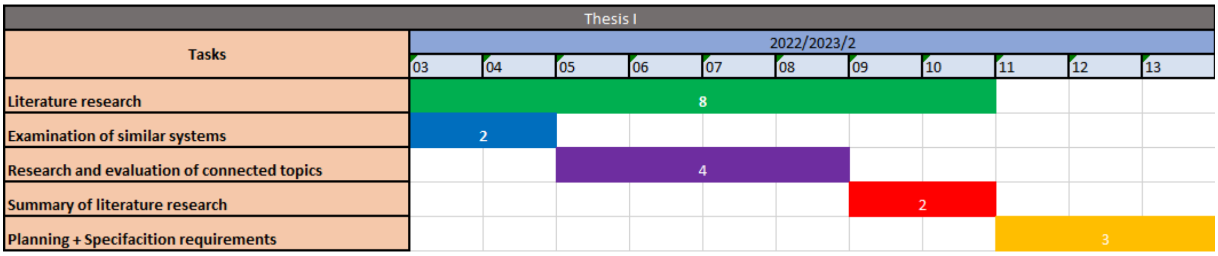


Figure 4.7 - Roadmap of Thesis I.

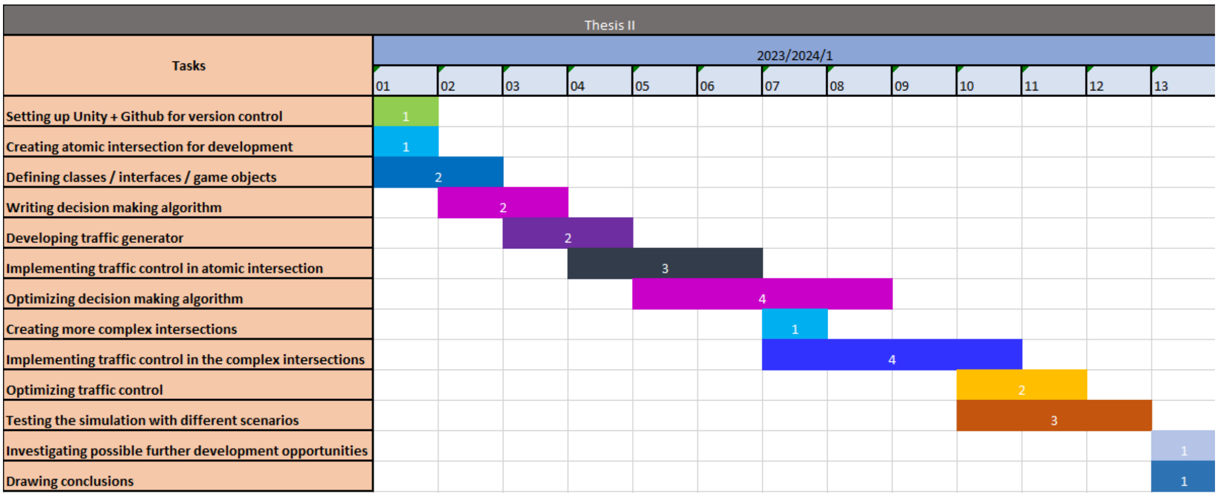


Figure 4.8 - Roadmap of Thesis II.

5. DEVELOPMENT

5.1. PARADIGM CHANGES DURING DEVELOPMENT

Road Ranking

During the development process I encountered a number of issues that led me to re-evaluate my approach to the realization of the project's goals. As discussed in my Thesis, I intended to use an explicit road rank system as suggested by the findings of my initial research into existing systems. However, this got proven unnecessary during the development process.
I slightly underestimated the impact of the real-time traffic data-ingest on the overall system performance in regards to the ability to react to drastic and subtle changes in traffic flow. Along the way I had to realize that the system performs well without the need to set up a preset road rank system. On the other hand a road-ranking approach can prove to be beneficial for the fine tuning of the system. By extending the inbound traffic detector's range a road can be promoted to become a pseudo-major road compared to the rest of the roads in the intersection. The reason for this is that by detecting more incoming vehicles from further away the detector feeds a larger BATCH (formerly VPM) value, which in turn forces the system to prioritize that given lane or road. Conversely, decreasing the range of the detector will cause the system to lower the observed road's relevance when calculating the length of the time-slots allowed for each green light.

Right-To-Control Mechanism

In my Thesis, I predicted that a central decision making hub can be avoided.
During development I tested my concept of a so-called Right-To-Control (RTC) mechanism which implements a "relay race' type of logic with each light receiving and passing on a "baton". This effectively promotes one light to act as a "conductor" for the entire node for a certain period of time. This also allows each traffic light to run the exact same logic code, making the system more modular and maintainable.
I was concerned that by breaking the OOP principle of one class not interfering with the operation of another class on the same hierarchical level I would introduce a non-viable solution, but I realized that by promoting one class to a control role would overcome this problem.

Cooldown Mechanism

While originally rejecting the idea of a turn based system, I came to realize that it is imperative for a certain control order to be established mainly to avoid the ping-pong effect of two lights passing the Right-To-Control to each other in an infinite loop. This was solved by introducing the cooldown mechanism that is a popular solution in gaming to avoid over-using the same effect repeatedly.

Detectors

Detectors provide real-time traffic data by counting passing vehicles and are the core components of any adaptive traffic control system. Although I did not specify the type of detectors to be used in real-life, in my simulation I needed to come up with a solution to be able to detect incoming and outgoing vehicles reliably. This resulted in a solution using Unity's built-in raycasting capabilities mimicking the behavior of a laser distance meter. Later I realized that this solution can be turned into an actual working physical detector that could be used in the physical world without the privacy considerations of camera/AI based systems. I will discuss this solution in the Further Development Avenues (Chapter 8.) section of my Thesis.

5.2. COMPONENTS OF THE SIMULATED SYSTEM

5.2.1. Overview

The following illustration [Figure 5.1] depicts how the PersistentScene relates to both the individual test scenes and their respective components.

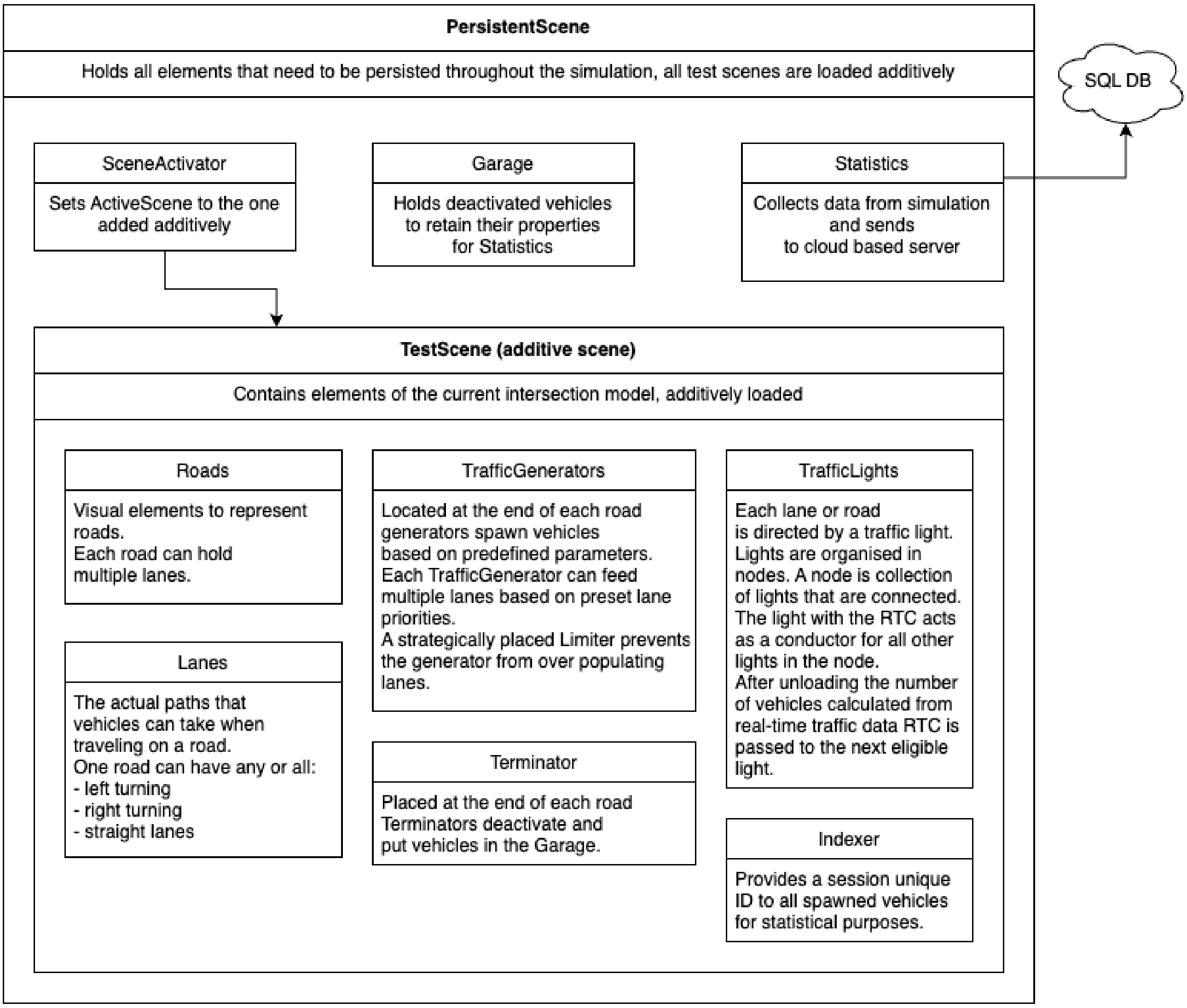


Figure 5.1 - Overview of Component Relations

5.2.2. Detailed System Components in the Simulation

- **Persistent Scene:** holds all elements that need to be persisted throughout the simulation, all test scenes are loaded additively
    - **Main Camera and Light**
    - **Compass:** Basic gizmo for easier orientation
    - **Statistics:** Collects data from the simulation and sends to cloud based server
    - **Garage:** Holds deactivated vehicles to retain their properties for Statistics
    - **Scene Activator:** Sets ActiveScene during runtime to the one added additively.
    - **Scanner:** the simulation of a theoretical solution for counting passing cars
- **Individual Test Scenes:** Contains elements of the current intersection model, additively loaded
    - **Intersection:** this zone covers the central (common) section of the intersection and is responsible for disabling collision detection inside the intersection to avoid cars accidentally stopping there
    - **Traffic Generators:** Located at the start of each road, generators spawn vehicles based on predefined parameters. Each TrafficGenerator can feed multiple lanes based on preset lane priorities. A strategically placed Limiter prevents the generator from over-populating lanes.
        - **Limiter:** Stops the TrafficGenerator from over-spawning the road.
    - **Indexer:** Provides a session-unique ID to all spawned vehicles for statistical purposes.
    - **Roads:** Visual elements to represent roads. Each road can hold multiple lanes.
        - **Terminator:**
            - deactivates and puts vehicles in the **Garage**
            - placed at the end of each road
    - **Lanes:** the actual path, consisting of waypoints that vehicles follow when traveling on a road. May lead straight or turn left or right
    - **Traffic Lights:** Each lane or road is directed by a traffic light. Lights are organised in nodes. A node is a collection of lights that are connected. The light with the RTC acts as a conductor for all other lights in the node. After unloading the number of vehicles calculated from real-time traffic data RTC is passed to the next eligible light.
        - **Blocker:** BoxCollider that when hit by a vehicle's BoxCollider stops the vehicle. Simulates red light. Green light is simulated by moving the blocker out of the vehicle's path.
        - **Inbound Detector:** Forward facing raycast that feeds real-time traffic data by counting the incoming vehicles. Provides one of the most important input data for the traffic light's RTC logic. This simulates a real time traffic sensor, as discussed in chapter **2.2.3.**

- **Outbound Detector:** Downward facing raycast that counts vehicles passing below. Can be placed at strategically relevant locations to properly count vehicles leaving the intersection or the area affected by the light it is connected to.

5.2.3. Traffic Light Logic Overview

The following flowchart [5.2] outlines the main logic behind an individual traffic light's decision making process. Each light runs the same logic, making a central decision making hub unnecessary as each light can act as the designated controller for the entire node.

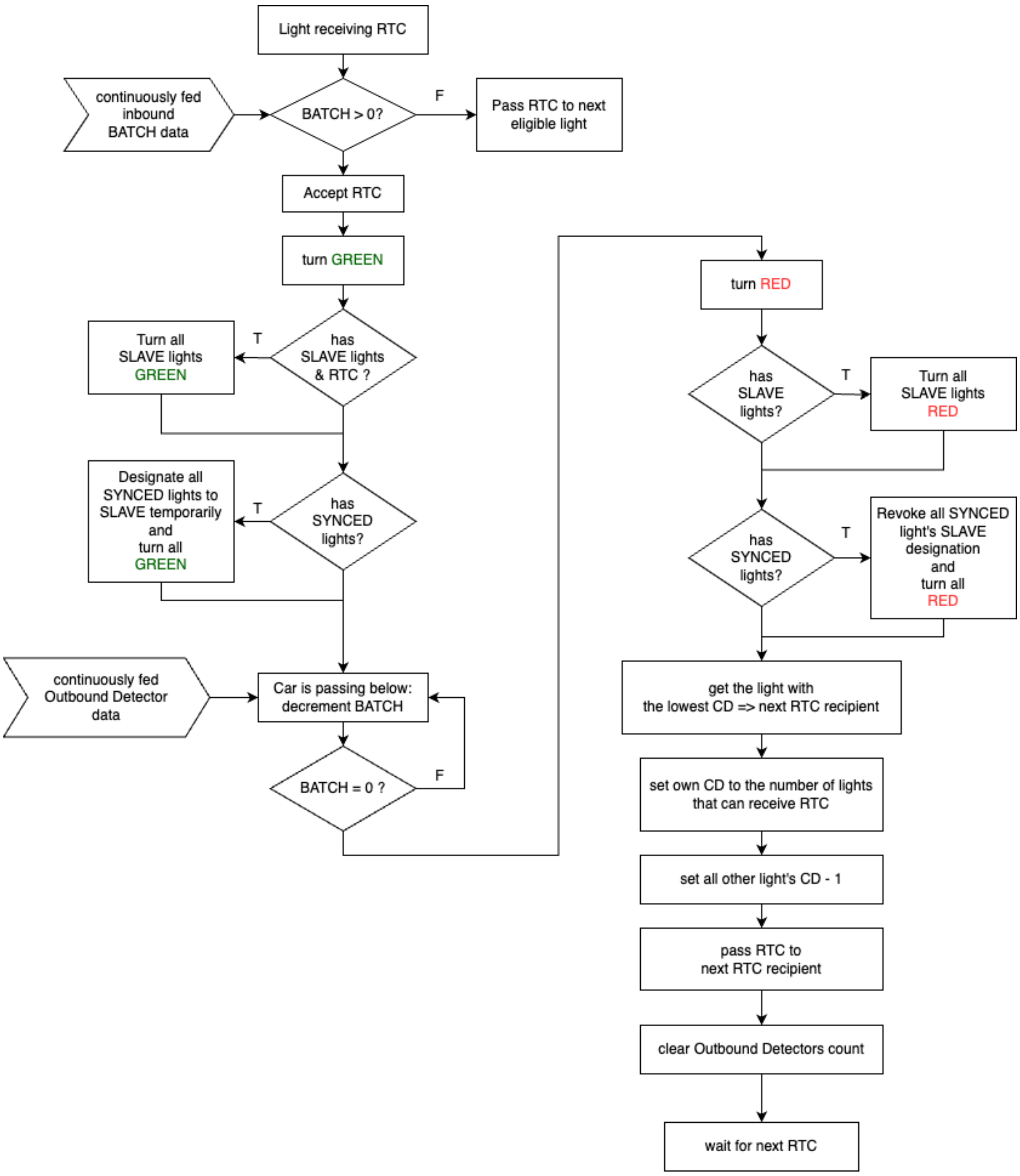


Figure 5.2 - Traffic Light Logic Flowchart

5.2.4. Explanation of Logical Elements

- **RTC:** The Right-To-Control designation received by and passed on by each light included in the decision making process in the node.
  While a light has the RTC it will:
  - act as a “conductor” towards all other SYNCED and SLAVE lights (see below), controlling their behavior
  - processes inbound and outbound traffic data
  - based on traffic information, decide when and to which other light to pass the RTC designation.

  Prior to a light accepting RTC a decision is made by the light to either accept or reject (pass on) designation based on real-time traffic data. If there are no vehicles waiting at the light, RTC must be passed on to the next light in the queue. See CD for further explanation.
  Slave lights are excluded from receiving RTC.
- **CD:** Stands for cooldown. Each light has a cooldown period. Opposed to other solutions, CD is not defined as a hard coded preset. Cooldown period depends on the time it takes all RTC capable lights to receive, accept (or reject) and process RTC once before circling back to the first light in the queue (one round). This prevents the occurrence of a ping-pong effect when two lights pass RTC between them in an infinite loop. Before a light passes RTC it decrements all other light’s CDs and sets its own CD so that it places it at the end of the queue.
- **BATCH:** The number of cars being processed by the light. Batch count is updated continuously based on detector data and evaluated when a light receives RTC to determine if the light can accept RTC (there are cars waiting) or reject and pass on RTC (no cars waiting). BATCH is then decremented each time a vehicle leaves the intersection. This is one of the main concepts behind this system's capability to react to changes in the flow of traffic.
  Let’s consider a situation when there’s an accident on one of the roads that entirely blocks traffic: the corresponding light will pass RTC on to the next eligible light when it detects that no vehicles are waiting thus freeing up time that the other lights can distribute among them.
- **SYNCED lights:** Any RTC capable light that will be directed by the light that currently has the RTC. A synced light differs from a SLAVE light (see below) in the term that it can receive RTC designation while a SLAVE light cannot.
- **SLAVE lights:** Slave lights are forced to turn green/red by a “master” light in the node. Slave lights cannot receive RTC, make decisions or control other light’s behavior in the node. Usually these lights control traffic on left turning lanes. Basically it is the same light as the “master” light, but controlling another lane.

5.3. Milestones of Development

5.3.1. Code Development

- Empty Unity project created, Persistent and additive scene loading implemented
- Basic road object and essential gizmos added to easily identify visual objects
  - Importing 3D model assets for road
- **Terminator object created**
- Test scene with a basic intersection created
- TrafficGenerator boilerplate created
- 3D car model and Prefab added (Blender FBX format import)
- **TrafficGenerator tested - success**
- Indexer created - Indexer keeps track of spawned cars and provides a unique id to each
- Lane prefab created
- **Pathrunner script created -** Testing follow path function for vehicle - **success**
- Detector created
- Skeleton **VehicleMotor added to Vehicle Prefab**
- TrafficGenerator now continuously spawns vehicles and assigns random lanes based on predetermined percentages (favor right turn, left turn or straight directions)
- Terminator now destroys cars at the end of their path - *this will be changed to deactivate only*
- Vehicle spawn interval can now be adjusted from the Inspector on individual TrafficGenerators for a more flexible simulation
- **Blocker added to TrafficLight** - to stop inbound vehicles
- Vehicle Collision detection added to VehicleMotor - vehicles now stop before collision
- Limiter added to TrafficGenerator to prevent the over spawning of lanes
- GUI controls for traffic lights added (STOP/GO) and traffic light Stop/Go function added - for testing purposes only
- Provisional road rank system implemented - *this will dropped in later iterations*
- **Intersection created**
- Garage added to Persistent Scene
- Terminator now deactivates Vehicles instead of destroying them and puts them in the Garage to retain their properties for statistical purposes
- **Minimally working intersection created**
  - All roads now have a TrafficGenerator that spawns vehicles with random paths and a TrafficLight that can allow or stop the flow of traffic
- Statistics skeleton built: for sending statistical data from Unity to the database
- Terminator now fires OnVehicleTerminated event with a Vehicle parameter for Statistics
- **Connecting to Microsoft Azure SQL server**
  - Added Microsoft.Data.SqlClient NuGet package to VS

    - changed Unity API compatibility level from .NET Standard 2.1 to .NET Framework
    - Connection successfully established to Azure SQL server
- **Standalone TrafficLight Detector skeleton created -** detector now counts incoming and stopped cars
- Implementing passed vehicle counter
    - Traffic light now counts the number of vehicles that passed underneath; will be used in smart traffic control logic
- Separated detector scripts from TrafficLight script to maintain modular approach
- **Outbound and inbound car detection is now done by separate detectors** - to allow multiple detectors on a single traffic light to collect data from multiple lanes
- **TrafficLight logic implemented**
    - Now each light is connected to the other lights
    - Right-To-Control (RTC) mechanism is in place *(further testing needed)*
    - OutboundDetector now notifies traffic light when a vehicle left
    - InboundDetector feeds real-time traffic data *(further testing needed)*
- Basic vehicle statistics implemented:
    - Travel time (total time spent on scene)
    - Idle time (total time spent waiting at intersection)
- **Implemented intersection statistics**
- SessionData statistics implemented
- **Model of Szentendre intersection created**
- Traffic Light logic improved (unloads roads more efficiently)
    - TrafficLights now count traffic until the light receives RTC again
    - Vehicle IDs now include TrafficGenerator IDs (for easier identification in database)
- New Scriptable Object created
    - SessionData to hold persistent data between sessions, such as session ID and other global constants
- Control logic improved
    - ChangeColor method created to make code DRY
    - Session Data filename changed to SessionData.cs
    - Control Logic now includes the option of slave and synced lights
- Visual improvements
    - Truck and Landspeeder models added
    - TrafficGenerator now randomly picks vehicle types
    - TrafficLight gizmos can now be turned off from SessionData
- Statistics now saves preliminary data to Azure Cloud Database
    - The following data are saved at the end of the lifetime of a vehicle:
      id, session_id, scene, vehicle, travel_time, idle_time
- First statistics created from data exported from Unity

- **Various intersections (M3, Basic Y, etc.) built for testing purposes**
- **Testing Sessions and Data Acquisition**

5.3.2. Non-code development

A collection of 3D models for traffic lights [Figure 5.3] and vehicles [Figure 5.4] used in the simulation, created and imported from Blender [28] in FBX format [29].

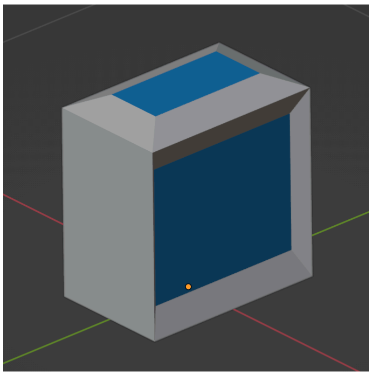


Figure 5.3 - TrafficLight model

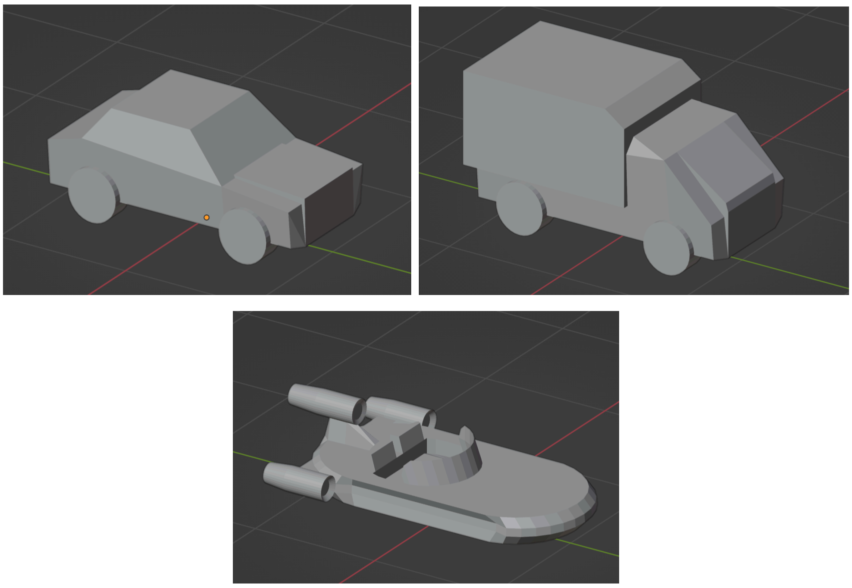


Figure 5.4 - Vehicle models

5.4. Simulated Intersections

The following four test scenes are created to explore how the proposed system behaves controlling the most common intersection types. Each scene will be set up to emulate different traffic scenarios, traffic load, time-of-day and emergencies.

5.4.1. Basic One-Lane Intersection

An imaginary intersection of two roads with similar traffic loads as the initial state. A combination of two Atomic X intersections. It is imperative that the unloading of the intersection be done in a manner where each road gets its own turn. This eliminates the blocking of the flow of traffic when vehicles need to turn to left or right while at the same time elevates the safety of the intersection by not allowing opposing traffic to cross paths.

Representation of a basic one-lane intersection in Unity[Figure 5.5].

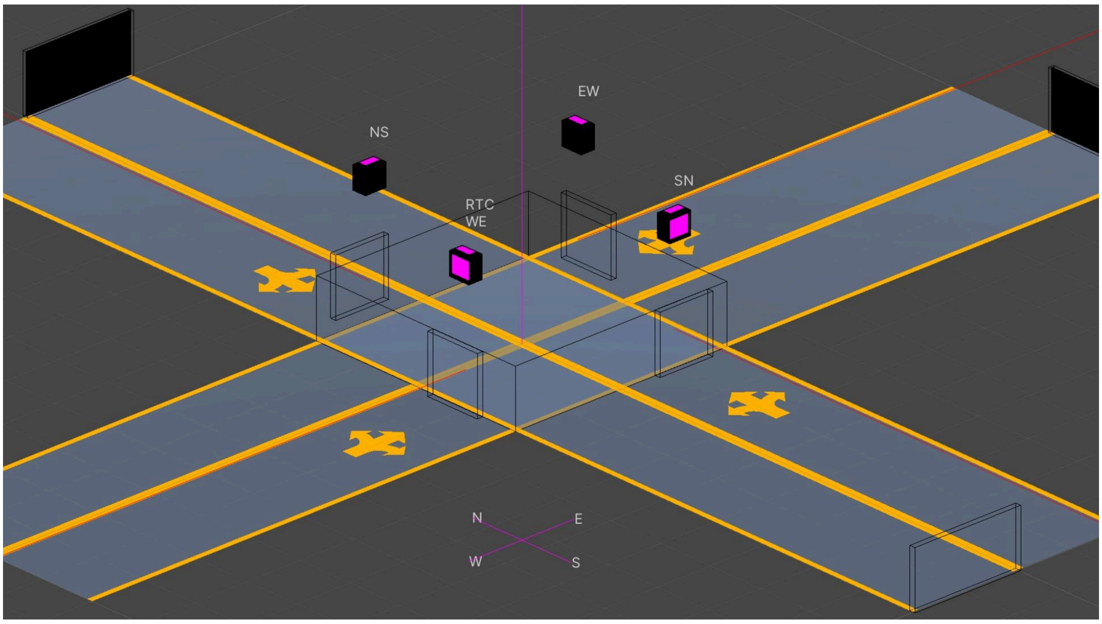


Figure 5.5 - Basic One-Lane Intersection in Unity

5.4.2. Szentendre

This intersection is one of the intersections I frequently use and shows best the drawbacks of current scripted traffic control systems. For this reason I picked this particular intersection to demonstrate the benefits of converting it to a smart traffic controlled crossroad. I have omitted the two right-turn lanes (North to West and West to South). They are irrelevant for the purpose of this simulation as they need not interfere with the control logic of the traffic lights.

Schematics of intersection at Szentendre and Dobogókői út [Figure 5.6] overlayed with real time traffic data from Google Maps.

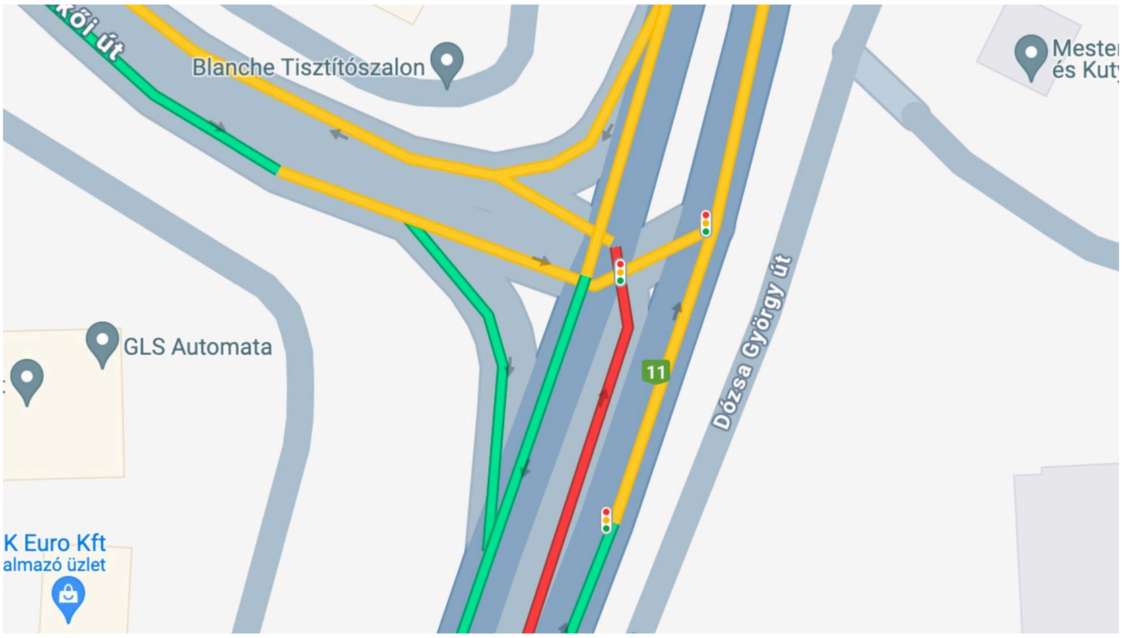


Figure 5.6 - Image of Szentendre and Dobogókői út from Google Maps

Representation of Szentendre and Dobogókői út in Unity[Figure 5.7].

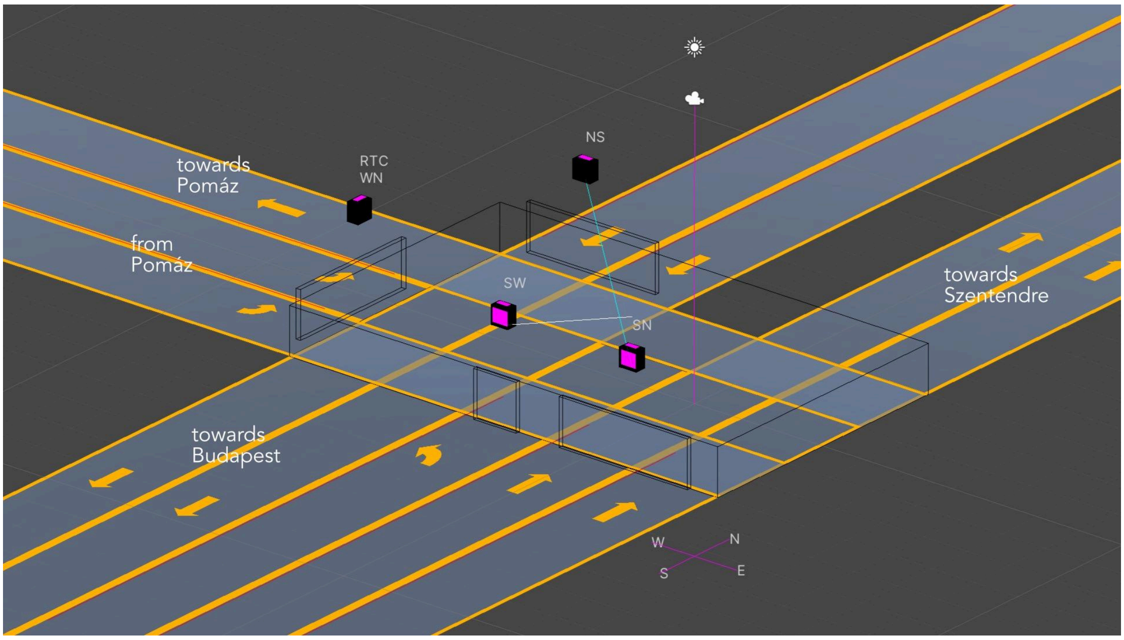


Figure 5.7 - Representation of Szentendre and Dobogókői út in Unity

5.4.3. M3

To test the behavior of my system in a large traffic volume arterial road versus a low traffic volume crossroad, I picked the M3-Szerencs utca. The large difference between the volume of traffic on these roads can stress test my system's capabilities even in extreme situations.

Schematics of the intersection at M3 junction at Szerencs u. [Figure 5.8] overlayed with real time traffic data from Google Maps and its simulation in Unity [Figure 5.9].

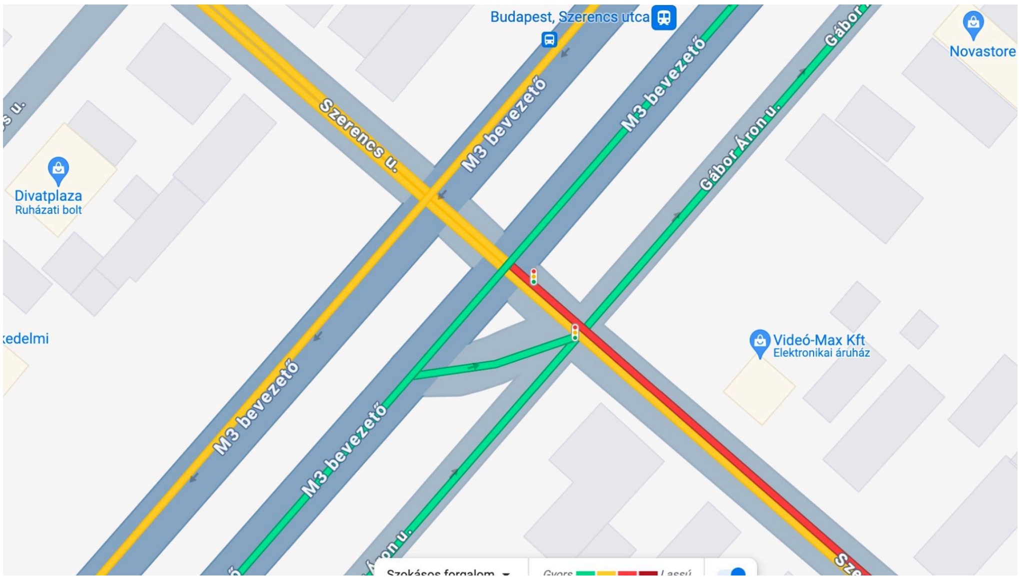


Figure 5.8 - Image of M3 junction at Szerencs u. from Google Maps

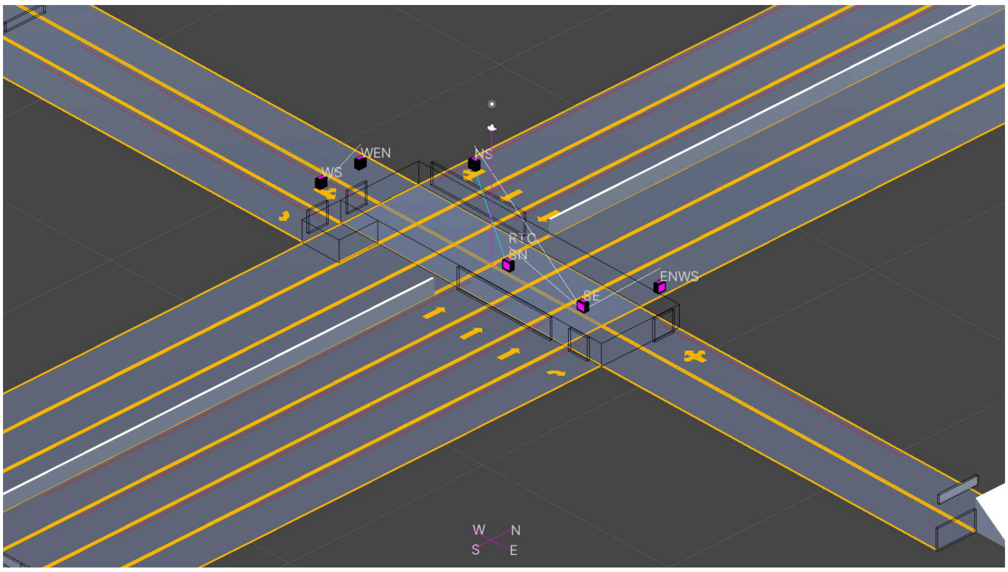


Figure 5.9 - Image of M3 junction at Szerencs u. in Unity

5.4.4. Basic Y

To test one of the simplest possible intersections, I assimilated this T intersection that is composed of three atomic Y intersections (see figure 4.2).

An imaginary intersection [Figure 5.10].

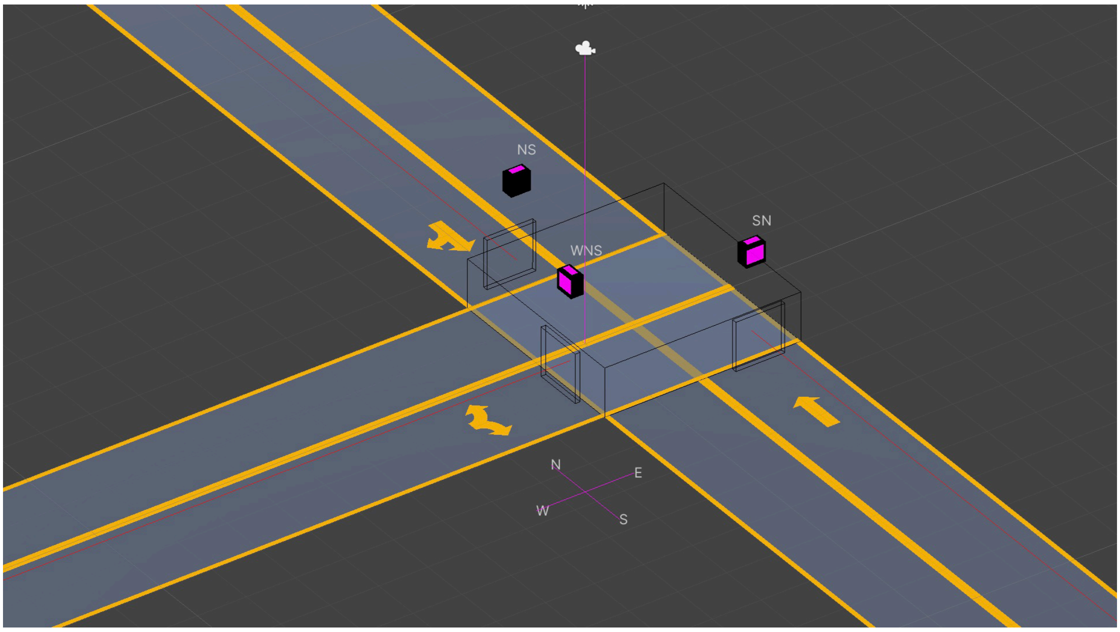


Figure 5.10 - Representation of a Basic Y intersection in Unity

6. TESTING

6.1. COLLECTED DATA AND METHODS

To provide digestible data, I introduced the following data groups and parameters to be collected and analyzed:

- Session Data:
    - *session_id*: unique integer that is incremented at every test run
    - *scene*: the name of the simulated intersection
    - *total_runtime*: total simulation runtime
- Vehicle Based Data (dbo.vehicles_data table):
    - *id*: AUTO INCREMENT
    - *session_id*: the unique id generated by the Session ScriptableObject and updated at each test session start
    - *scene*: the name of the simulated intersection
    - *vehicle_id*: the unique id of a spawned vehicle provided by the Indexer script
    - *travel_time*: the timespan from the creation of the vehicle instance by the TrafficGenerator to its deactivation by the Terminator
    - *idle_time*: the total time the vehicle spent waiting at red lights
- Intersection Based Data:
    - *id*: AUTO INCREMENT
    - *session_id*: unique integer that is incremented at every test run
    - *scene*: the name of the simulated intersection
    - *intersection_empty_timespan*: clocked when there are no vehicles in the Intersection
    - *passing_vehicles_count*: the number of vehicles entering and exiting the Intersection

All statistical data is relayed to the Statistics component via events triggered by the class responsible for that particular event. The vehicle script for example would pass the Vehicle object to the Statistics class with all the relevant data accessible in public fields when the object gets deactivated and parented to the Garage. The Garage component's sole purpose is to retain all properties of a Vehicle instance until the simulation is terminated.

6.2. DATA COLLECTION PIPELINE CONSIDERATIONS

I choose an event-driven data collection pipeline that inserts and/or updates data in the database when it is produced by the simulation. The reason for this approach is that even if a

simulation fails at some point the data collected up to that moment would be preserved. This gives a valuable insight into the cause of the failure.
I observed that the Unity Engine is prone to mishandling collisions from time to time causing the vehicles to get stuck. These occurrences are outside of my control and happen for no apparent reason. This prompted me to drop my initial approach of collecting data locally and upload to the cloud server when the simulation ends. Instead, I opted for an asynchronous approach that can store statistical data as they happen. This solution works seamlessly and produces the expected results.

To evaluate and compare the effectiveness of any traffic control system, I propose using the ratio between the *total_runtime* (TR) and the *intersection_empty_timespan* (IET) ratio named Intersection Empty Ratio (IER). This number must approach 0 to prove that an intersection is being utilized at optimal capacity. Since I do not have access to real-life data it is currently not possible to compare the simulations' performance to their real-life counterparts. This comparison needs further resources and readily available statistical data to be completed. However, comparing the TR vs IET ratios of a series of simulations of the same intersection, but with different traffic loads will yield a proper base for comparison.

6.3. EXPECTED BEHAVIOR

1. Vehicles must not collide inside an Intersection.
Vehicle behavior will be observed visually and the simulation producing a collision inside an Intersection will be terminated and marked as a failed test. Vehicles leaving the intersection no longer affect the outcome of the simulation and their collisions will be ignored.

2. Roads with heavier traffic loads should get more "green time".
This will be proven by the data produced by the simulation. A comparison will be made between the traffic load and the VIT (time spent waiting at the red light).

3. The intersection's empty time (IET) should not be affected significantly by any traffic scenario change other than having not enough vehicles to saturate the intersection due to low traffic volume. In the case of a road or a lane getting completely blocked (accident) the corresponding light should not receive RTC designation until there are no cars present at the light thus the IET is expected to decrease as the RTC queue gets shorter.

6.4. TESTING SCENARIOS

Each test is run to include 100 vehicles and multiple light cycles. Note that all simulations use the same control logic on each light. Testing scenarios are set up with an initial state that is to be adjusted by the control logic to best fit the traffic flow needs of each intersection.

It is considered high traffic when the volume is at least double of the low traffic load's volume.
I have tested the following four different scenarios each with its unique parameters:

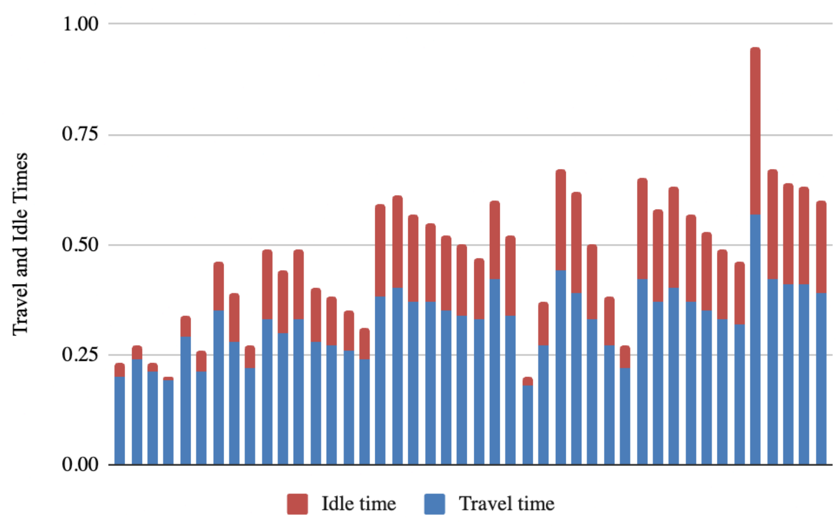
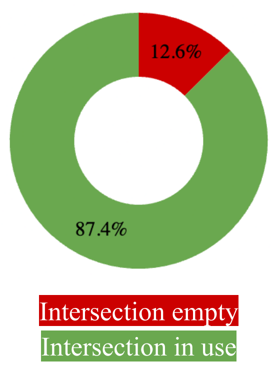
- Basic One Lane
- Szentendre-Dobogókői út intersection
- Basic Y intersection
- M3-Szerencs utca intersection

After each test scenario pair their respective test results can be seen.

6.5. TEST SCENARIOS - DETAILED SETUP AND TEST RESULTS

Test Scenarios for simulated intersection 5.3.1. (Basic One Lane):

- Test scenario 5.3.1/a:
  - high traffic on SN and EW roads
  - low traffic on NS and WE roads
  - no interruption to the flow of traffic
- Test scenario 5.3.1/b:
  - high traffic on SN and EW roads
  - low traffic on NS and WE roads
  - accident (blocking road entirely) on EW high traffic road

<table>
<tr>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>Average Travel Time (ATT): 30.2s
Average Idle Time (AIT): 14.7s</td>
<td>IER: 0.14
TR: 150s</td>
</tr>
</table>

(Cont'd) **Conclusion**: While the AIT comprises approximately half of the ATT, the low IER value suggests that the intersection was optimally saturated.

Table 6.1 - Test Results of Test Scenario 5.3.1/a

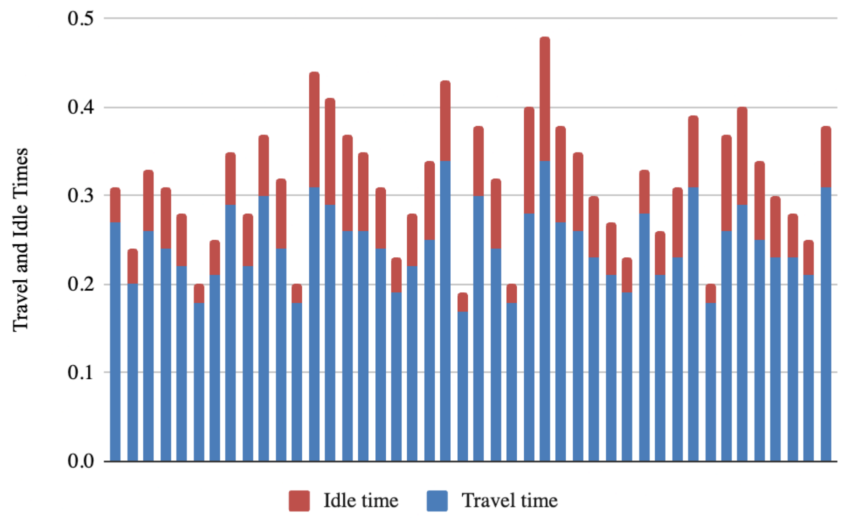
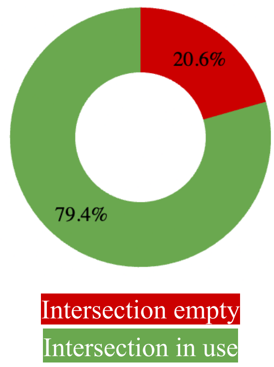
<table>
<tr>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>Average Travel Time (ATT): 21.6s
Average Idle Time (AIT): 6s</td>
<td>IER: 0.26
TR: 170s</td>
</tr>
<tr>
<td colspan="2">Conclusion:
1. The ATT and AIT values decreased significantly compared to 5.3.1/a. This is due to the smart system allocating more green-light time to the remaining roads. This is the expected behavior.
2. Because the road with the higher traffic load got blocked the intersection did not receive enough vehicles to produce a higher IER. This is the expected behavior.</td>
</tr>
</table>

Table 6.2 - Test Results of Test Scenario 5.3.1/b

Test Scenarios for simulated intersection 5.3.2. (Szentendre):

- Test scenario 5.3.2/a:
    - morning rush hour situation
    - high traffic on NS and WN roads
    - low traffic on SN and SW roads
    - no interruption to the flow of traffic
- Test scenario 5.3.2/b:
    - evening rush hour situation
    - high traffic on SN and SW roads
    - low traffic on NS and WN roads
    - no interruption to the flow of traffic

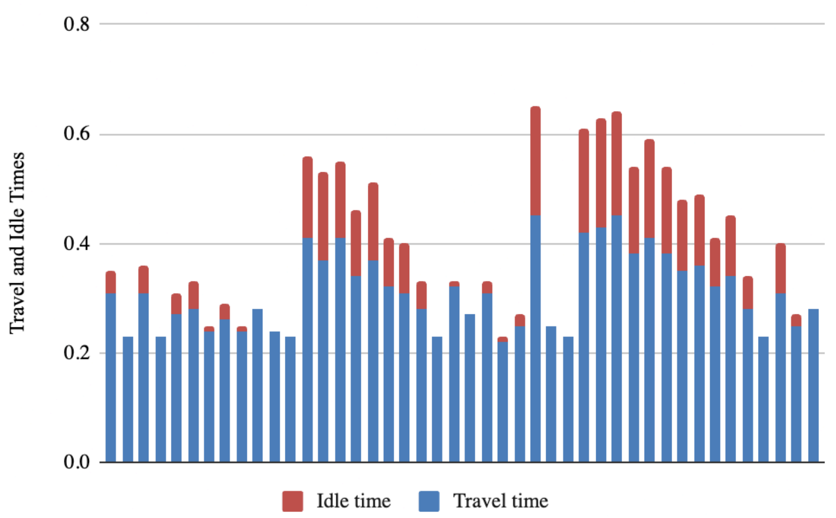
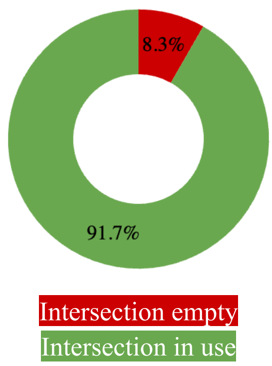
<table>
<tr>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>Average Travel Time (ATT): 31.1s
Average Idle Time (AIT): 10.3s</td>
<td>IER: 0.09
TR: 109s</td>
</tr>
<tr>
<td colspan="2">Conclusion: In the morning rush-hour the bottleneck of the 2 lanes leading towards the city center causes the intersection's IER to increase compared to the evening rush-hour times.</td>
</tr>
</table>

Table 6.3 - Test Results of Test Scenario 5.3.2/a

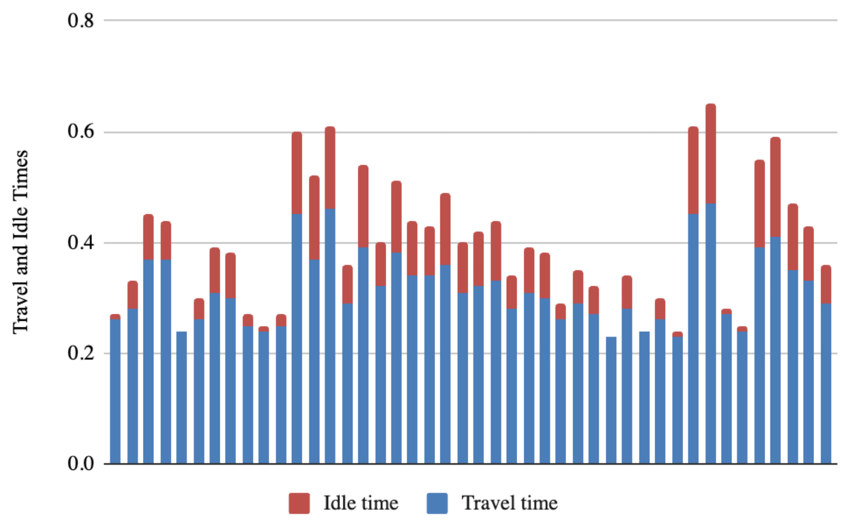
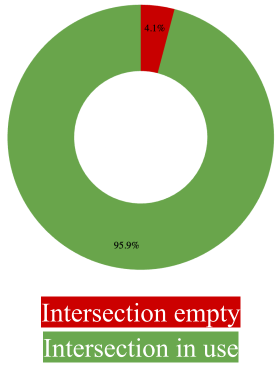
	
Average Travel Time (ATT): 29.4s Average Idle Time (AIT): 8.64s	IER: 0.04 TR: 98s
Conclusion: During the evening rush-hour with heavier traffic from the city center the saturation of the intersection improves as traffic flows on 3 lanes instead of 2 from the opposite direction.	

Table 6.4 - Test Results of Test Scenario 5.3.2/b

Test Scenarios for simulated intersection 5.3.3. (M3):

- Test scenario 5.3.3/a:
  - evening rush hour situation
  - high traffic on NS and SN roads
  - low traffic on WNES and ENWS roads
  - no interruption to the flow of traffic
- Test scenario 5.3.3/b:
  - morning rush hour situation
  - high traffic on NS and ENWS roads
  - low traffic SN road
  - road closure on WNES high traffic road

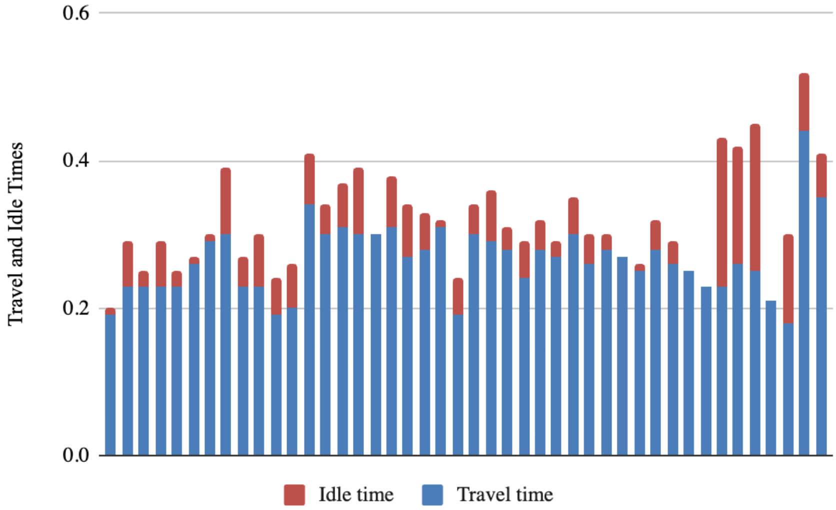
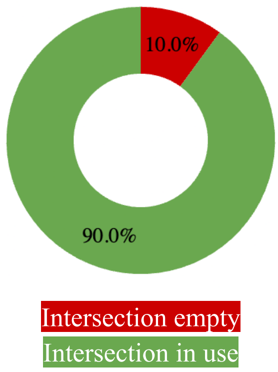
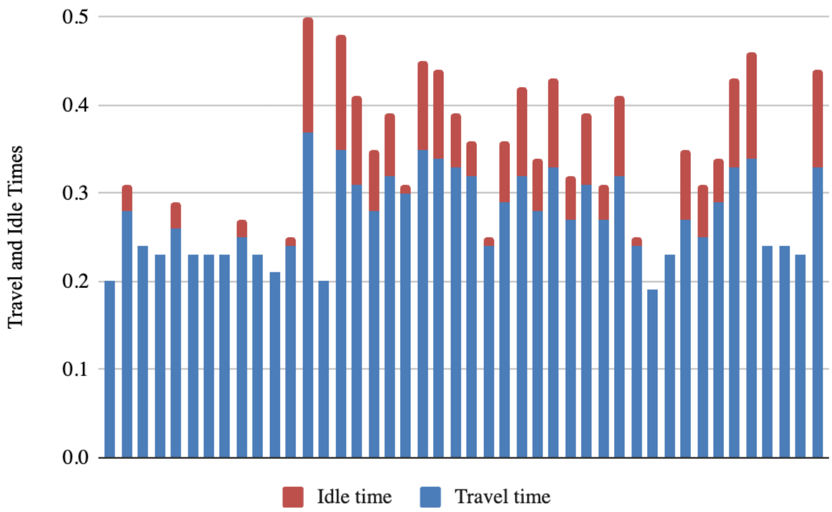
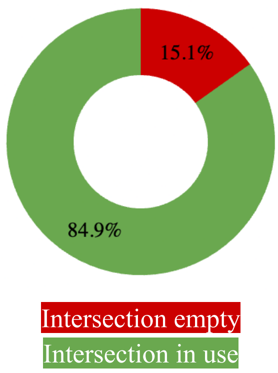
<table>
<tr>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>Average Travel Time (ATT): 25s
Average Idle Time (AIT): 6.9s</td>
<td>IER: 0.11
TR: 60s</td>
</tr>
<tr>
<td colspan="2">Conclusion: Baseline for a normal traffic flow on a multi-lane motorway/street intersection. Despite the densely populated lanes, a noteworthy distinction between AIT and ATT is evident, alongside an optimal IER, signifying smooth traffic flow even in a busy traffic scenario.</td>
</tr>
</table>

Table 6.5 - Test Results of Test Scenario 5.3.3/a

<table>
<tr>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>Average Travel Time (ATT): 25s
Average Idle Time (AIT): 5.18s</td>
<td>IER: 0.18
TR: 86s</td>
</tr>
</table>

(Cont'd) **Conclusion**: In comparison to 5.3.3/a, there is an even lower AIT observed, attributed to the closure of a typically high-traffic road. Despite this, the IER yielded an optimal outcome, suggesting the system's adaptive capabilities. It reacted by refraining from assigning the RTC and instead allocated more green time to the other traffic lights. This is the expected behavior of the system.

Table 6.6 - Test Results of Test Scenario 5.3.3/b

Test Scenarios for simulated intersection 5.3.4.:

- Test scenario 5.3.4/a:
  - normal, off-peak hours situation
  - medium traffic on NS, SN and WNS roads
  - no interruption to the flow of traffic
- Test scenario 5.3.4/b:
  - night time off-peak hours situation
  - low traffic on NS, SN and WNS roads
  - no interruption to the flow of traffic

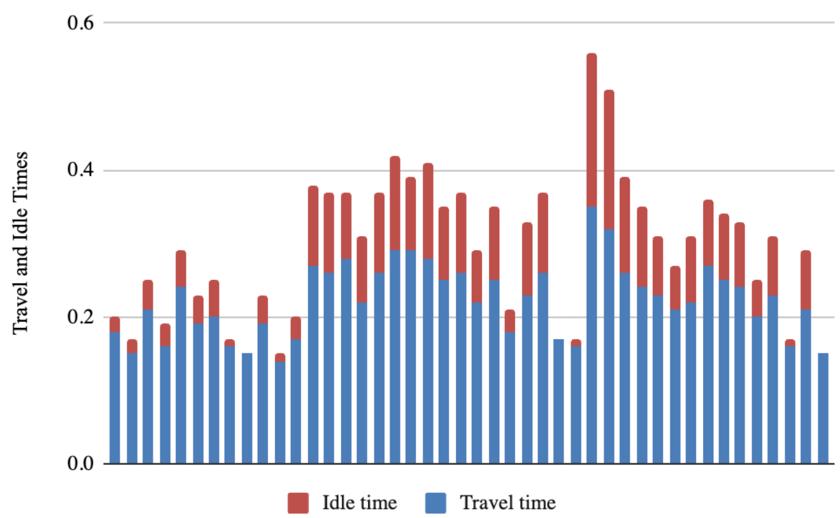
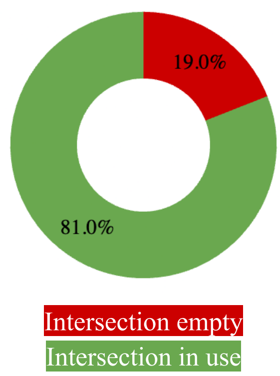
<table>
<tr><td>
</td><td>
</td></tr>
<tr><td>Average Travel Time (ATT): 20s
Average Idle Time (AIT): 6.9s</td><td>IER: 0.23
TR: 142s</td></tr>
<tr><td colspan="2">Conclusion: The 3-to-1 ratio of ATT to AIT implies the intersection operated optimally for a moderate traffic volume. This is further corroborated by the slight increase in the IER.</td></tr>
</table>

Table 6.7 - Test Results of Test Scenario 5.3.4/a

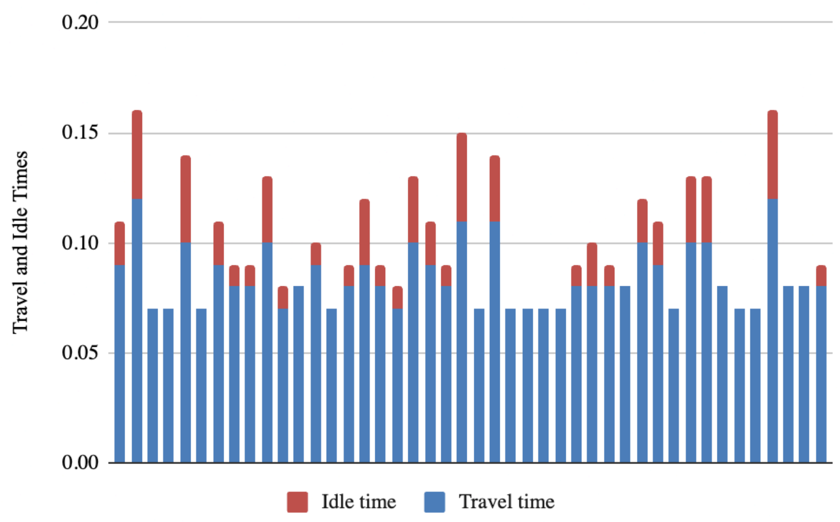
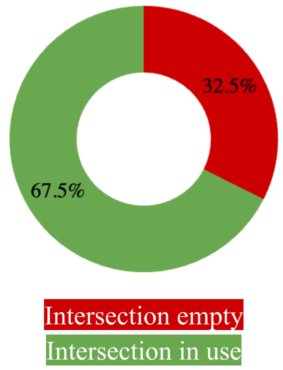
<table>
<tr>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>Average Travel Time (ATT): 7.8s
Average Idle Time (AIT): 0.86s</td>
<td>IER: 0.48
TR: 151s</td>
</tr>
<tr>
<td colspan="2">Conclusion: Due to the low volume of night time this traffic scenario did not produce enough vehicles for the intersection to reach its optimal saturation, however the system still produced the expected results as the idle time was kept at the lowest possible. This is the expected behavior.</td>
</tr>
</table>

Table 6.8 - Test Results of Test Scenario 5.3.4/b

7. EVALUATION

After concluding extensive testing sessions, I maintain that my approach is a valid alternative to the existing scripted/timed and the more complicated smart traffic control systems. It performs best when all nodes in a wider area (i.e. entire town) is controlled by its uniform logic.

According to the data collected and analyzed this system outperforms scripted systems in the following sense:

- It instantly reacts to changes in traffic flow patterns such as morning rush hour (vehicles moving towards the city) or evening rush hour (traffic flows to the opposite direction in greater value).
- Reacts to and handles unexpected traffic flow changes such as accidents or road closures eliminating unnecessarily designated green lights to roads with no traffic
- No need for preset values to be input for the system to reach a self-adjusting/self-balancing state. Initialization requires the setting up of lights, detectors and the connecting (wireless) network only.
- No need to promote any road or lane to higher priority. Detector components of the system can ensure that roads with larger traffic volume get more time to unload vehicles.
- No need to collect traffic data prior to initializing the system thus avoiding the need to rely on outdated traffic data to control the system.
- A central command hub can be omitted or used only to collect traffic data.
- Maintains driver/passenger privacy by not relying on video surveillance and image recognition.
- The lack of video surveillance makes the maintenance of large scale data processing unnecessary.

The system meets my expectations and successfully achieves its ultimate objective of eliminating unnecessary vehicle wait times at intersections in the absence of opposing traffic. It not only optimizes the utilization of road infrastructure but also minimizes environmental impact. In doing so, it proves to be an efficient and environmentally conscious solution, showcasing its effectiveness in enhancing traffic flow while demonstrating a commitment to sustainable practices.

8. FURTHER DEVELOPMENT AVENUES

During software development I emulated a real life vehicle detector using Unity's built-in raycasting capabilities that led me to the realization that this can be turned into an inexpensive laser distance meter based detector for real-life application.
Placed at strategic locations such as ingress and egress points of the intersection the downward facing detectors could continuously scan the shape of vehicles passing below. Such detectors would be capable of distinguishing vehicle types by analyzing the shape of the scanned curve. Since my system requires nothing more than the number of vehicles approaching and leaving the intersection, this would be an optimal solution.
Advantages of such system are:

- inexpensive, readily available technology (off-the-shelf laser measuring tools)
- can be mounted on existing infrastructure such as light posts and traffic light consoles
- all-weather solution
- no need for expensive and time consuming road works to be carried out
- privacy considerations are non-existent compared to camera/AI based systems

Exploring the real-life adaptation methods

As a fallback measure, my system might need to implement the functionality of being able to switch back to a timed/scripted operation. This could also be useful in the case of a detector failure. And also might be necessary to switch an entire town's current non-intelligent traffic control infrastructure to use this system. The process of switching from the old system to the new would involve first switching to the new system's scripted control and then switching to the new system's smart control in one go.

Emergency services detection

Integrating an emergency services detection system to force nodes to give way to emergency or high-priority vehicles. This calls for receiving a broadcast signal from the emergency vehicle that overrides the default logic of the node. I'm confident that this functionality can be easily implemented without heavily impacting established protocols.

Pedestrian traffic consideration

The current system does not account for pedestrian traffic, as it falls outside the scope of this thesis. While it's true that some real-world intersections may lack pedestrian crossing points (for instance intersections discussed in 5.4.2. and 5.4.3.), integrating the capability to consider pedestrian traffic is essential for a thorough transition in a city's overall traffic control infrastructure. One potential strategy involves deploying video cameras equipped with intelligent image recognition software. This technology could impact the functioning of the RTC logic by factoring in pedestrian traffic.

9. CONCLUSION

In my Thesis, I have set a goal to offer an alternative to scripted/timed traffic light control systems while employing the least complicated method to achieve a viable solution.

Due to the restricted extent and narrow scope of this paper, I feel that further testing and fine-tuning is necessary to utilise the full potential of such system with focus on testing more complex intersection connections, the integration of manual override for unforeseeable events such as natural disasters and the improvement of overall system performance.

A full-featured independent traffic data collection method could also be implemented based on sensor data provided by the system in connection with a more thorough analysis of the collected data by data-analytics experts.

I believe that the proposed solution in this paper is firm enough to build upon and encapsulates all the methods and procedures needed for a real-life experimental setup when explored further.

Having my goals to reduce congestion, driver frustration, saving fuel, reducing environmental impact and improve safety at intersections, I believe that my system - when implemented properly - could potentially provide these benefits in a cost-effective way.

BIBLIOGRAPHY

[1] 1868-2019: a Brief History of Traffic Lights (https://www.inclusivecitymaker.com/1868-2019-a-brief-history-of-traffic-lights/), last viewed: 2023. 04.17.

[2] A Summary of Vehicle Detection and Surveillance Technologies use in Intelligent Transportation Systems (https://www.fhwa.dot.gov/policyinformation/pubs/vdstits2007/05pt2.cfm), last viewed: 2023. 04.21.

[3] Econolite (https://www.econolite.com), last viewed: 2023. 03.22.

[4] Econolite Centracs® Mobility Supports Crowdsourced Traffic Data Analytics (https://www.econolite.com/econolite-centracs-mobility-supports-crowdsourced-traffic-data-analytics/), last viewed: 2023. 03.22.

[5] Fleck, J. L., Yee, B. M.: Safety evaluation of right turn on red. ITE Journal, Vol. 72., 2002, pp. 46-48

[6] Goodwin, P., Wright, G.: Decision Analysis for Management Judgment. Wiley, 2014

[7] Greenhouse Gas Emissions from a Typical Passenger Vehicle (https://www.epa.gov/greenvehicles/greenhouse-gas-emissions-typical-passenger-vehicle#typical-passenger), last viewed: 2023. 04.20.

[8] How Red-light Cameras Work (https://auto.howstuffworks.com/car-driving-safety/safety-regulatory-devices/red-light-camera.htm), last viewed: 2023. 04.21.

[9] How Traffic Lights Detect Cars Are Waiting for the Light to Change (https://auto.howstuffworks.com/car-driving-safety/safety-regulatory-devices/how-does-a-traffic-light-detect-that-a-car-has-pulled-up-and-is-waiting-for-the-light-to-change.htm?srch_tag=mqur664f3uslgtq7i3zwhuqkcxibblve), last viewed: 2023. 04.21.

[10] Infrared – Infrared Radiation (https://www.infratec.eu/sensor-division/service-support/glossary/infrared-radiation/), last viewed: 2023. 04.21.

[11] Lay, M. G. : Ways of the World: A History of the World's Roads and of the Vehicles That Used Them. Rutgers University Press, 1992

[12] Levinson, D. et al.: Traffic Signals (https://eng.libretexts.org/Bookshelves/Civil_Engineering/Fundamentals_of_Transportation/06%3A_Traffice_Control/6.02%3A_Traffic_Signals), last viewed: 2023. 04.16.

[13] Mircea, M., Nicolae, F.: Increasing of the urban traffic surveillance by automatic information devices. Open Engineering, Vol. 4., 2014, pp. 133-141

[14] Red light running defined (https://www.iihs.org/topics/red-light-running#red-light-running-defined), last viewed: 2023. 04.20.

[15] Red-Light Running Fatality Map 2019 (https://ncsrsafety.org/stop-on-red/red-light-running-fatality-map-2019/), last viewed: 2023. 04.20.

[16] Road Traffic Controller (https://www.yourcareer.gov.au/occupations/899923/road-traffic-controller), last viewed: 2023. 04.15.

[17] Rowe, E.: The Los-Angeles Automated Traffic Surveillance and Control (ATSAC) System. IEEE Transactions on Vehicular Technology, Vol. 40., 1991, pp. 16-20.

[18] Sitraffic Traffic Center Systems (https://assets.new.siemens.com/siemens/assets/api/uuid:5b21b17c86faa4bb6c3b2766d2f4c303f46abdf3/sitraffic-centrals-v2-en-180120.pdf), last viewed: 2023. 03.22.

[19] Sources of Greenhouse Gas Emissions (https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions), last viewed: 2023. 04.20.

[20] SynchroGreen Real-Time Adaptive Traffic Control (https://www.cubic.com/sites/default/files/2019-08/synchrogreen_brochure.pdf), last viewed: 2023. 03.24.

[21] The history and evolution of traffic lights (https://elteccorp.com/news/other/the-history-and-evolution-of-traffic-lights/), last viewed: 2023. 04.16.

[22] The History and Meaning of Colored Traffic Lights
(https://www.idrivesafely.com/defensive-driving/trending/history-and-meaning-colored-traffic-lights), last viewed: 2023.04.17.

[23] Traffic Control in the Smart City Using Video Cameras and Analytics
(https://www.intelli-vision.com/2019/08/30/traffic-control-in-the-smart-city-using-video-cameras-and-analytics/), last viewed: 2023. 04.21.

[24] Traffic signals
(https://wsdot.wa.gov/travel/operations-services/traffic-signals), last viewed: 2023. 04.16.

[25] Unity
(https://unity.com), last viewed: 2023. 05.06.

[26] Unreal Engine 5
(https://www.unrealengine.com/en-US/unreal-engine-5), last viewed: 2023. 05.06.

[27] What is Unity?
(https://gamedevacademy.org/what-is-unity/#What_is_Unity), last viewed: 2023. 05.06.

[28] Blender
(https://www.blender.org/), last viewed: 2023. 11.20.

[29] FBX File Format Specification
(https://code.blender.org/2013/08/fbx-binary-file-format-specification/), last viewed: 2023. 11.20.

LIST OF FIGURES AND TABLES