

VERSION 4.6

16/01/2022



DYNAMIC TRANSPORTATION LAWS IN A SMART CITY

SIMULATOR TO OPTIMIZE THE FLOW OF TRAFFIC FOR AUTONOMOUS
VEHICLES, BY DYNAMICALLY CHANGING TRAFFIC LAWS AND
LEVERAGING THE USAGE OF ROADS AND INFRASTRUCTURE

PROJECT CODE: 21220340

PREPARING REPORT – FINAL PROJECT 2022

ACADEMIC INSTRUCTOR: MR. YORAM SEGAL

PRESENTORS

STUDENT NAME 1: 312272230, SAAR ELIAS

STUDENT NAME 2: 312389083, NAOR YEKUTIELY

INSTRUCTORS SIGNATURE



תוכן עניינים

1 SUMMARY	2
1.1 תקציר בעברית	2
1.2 summary	2
1.3 keywords list	2
2 Tables and terms	3
2.1 Table of figures	3
2.2 Talble of tables	3
2.3 Acronym	3
3 introduction	4
3.1 defining the problem	4
3.2 The technological CHALLENGE	4
4 Ways to solve the problem	5
5 expected project artifact	6
6 Describing a similar idea that can be inspiring	6
7 Risks, uncertainty and project constraints	6
8 similar works in the field	7
9 Functional requirements	8
9.1 Project requirements	8
9.2 project operation and usage	8
9.3 system requirements	9
10 What has been done so far in the project	9
11 planning	14
11.1 Milestones table	14
11.2 Tasks table	15
11.3 Gantt	17
12 References	0
appendices list	1
Appendix A – block diagram	2



1 SUMMARY

1.1 תקציר בעברית

בפרויקט זה אנו מציגים גישה חדשנית בתחום פתרונות התחבורה בערים חכמות ועתידניות. עם הצמיחה המשמעותית של תחומים כגון: רכבים האוטונומיים, רשתות 5G, יכולות עיבוד גבוהות בענן העתיד של תחום התחבורה בתוך ערים חכמות צפוי להשתפר משמעותית. בעיר עתידנית שבה כלל הרכבים הם אוטונומיים לחלוטין ואין שיקול דעת אנושי בהחלטות המתרחשות כחלק מזרימת התנועה, אנו מציגים צורת חשיבה חדשה בנוגע לשימוש בתשתיות התחבורה בערים שכאלו. כל חוקי התנועה יהפכו לדיגיטליים והם יהיו דינמיים לחלוטין. שינויים דרמטיים אילו, פותחים דלתות חדשות בנוגע לדרכים לניצול מרבי של תשתיות התנועה וייעול של זרימתן. בפרויקט זה אנו מציגים עיר חכמה שכזו, על ידי מידול שלה לידי סימולטור תנועה שבו חוקי התנועה משתנים באופן דינמי. בנוסף, כחלק מהפרויקט אנו נציג ניתוח של המידע שהסימולציה מייצרת לצד השוואה בין הארכיטקטורה החדשנית הזו, לבין המצב הנוכחי בעזרת טכנולוגיית הצגת וניתוח מידע. במסגרת פרויקט זה אנו לא מכוונים להציג אלגוריתמים של אופטימיזציות תנועה, אלא להשתמש באלגוריתמים קיימים שכאלו על מנת להראות את נכונות הנחת היסוד שלנו. אנו מעודדים אחרים לקחת השראה מהפרויקט ומהרעיון היסודי שלו ולשפר אותו, או לממש את הרעיון של שינוי חוקי תנועה דינמיים בעיר חכמה בצורה שונה לחלוטין. לפרויקט זה, קיים נתבך סביבתי שכן הפרויקט מכוון להפחתת הזיהום הנגרם על ידי פליטות Co2 בערים אורבניות מסביב לעולם.

1.2 SUMMARY

In this project we present an innovative approach towards transportation solution in a futuristic smart city. With the rising of new technologies such as Autonomous Vehicles (AV), 5G networks, and cloud computing the future of transportation within smart cities is expected to improve in many factors. In a futuristic smart city, when transportation will be autonomous and no human decision will take part, we present a new way of thinking about transportation and road infrastructure in general. All traffic laws (TL) will be digitalized and dynamic. These dramatic changes will open up new ways for optimizing traffic and using the road infrastructure in the most efficient way possible. This project presents an architecture for such futuristic system as well as intuitive examples for how such system can be implemented. The implementation we present in this project is a traffic simulator that utilizes all the benefits of having dynamic traffic laws. The simulator presents a futuristic city where all the vehicles are autonomous and the traffic laws change dynamically. The project also includes data analysis and comparison between this new approach against the current one in an interactive and sophisticated dashboard using data presentation technologies. In this project we do not aim to present a certain optimization technic but rather to show the possibility for such system existence in the future, and encourage others to implement traffic optimization algorithms with this new futuristic approach. Furthermore, this project is eco-friendly as it aims to reduce the carbon footprint within urban cities around the world.

1.3 KEYWORDS LIST

Dynamic traffic laws, Traffic simulator, Smart city, Autonomous vehicle, Python Pygame project, Machine learning, carbon footprint



2 TABLES AND TERMS

2.1 TABLE OF FIGURES

Figure 1: ONE WAY TRAFFIC CONGESTION.....	4
Figure 2: the microscopic ts model approach	7
Figure 3: system implementation	8
Figure 4: example of the base simulator	11
Figure 5: driver license categories	12
Figure 6: distinct vehicles	12
Figure 7: simulation metrices	13
Figure 8: pixelized map.....	13

2.2 TALBLE OF TABLES

Table 1: Milestone table.....	14
Table 2: Tasks table	15

2.3 ACRONYM

Dynamic Transport Law System(DTLS) – Abstract system concept which utilizes fully automated vehicles traffic flow, with dynamic traffic laws.

Autonomous Vehicle(AV) - A ground vehicle that is capable of sensing its environment and moving safely with no human input.

traffic simulator(TS) - The mathematical modeling of transportation systems (e.g., freeway junctions, arterial routes, roundabouts, downtown grid systems, etc.) through the application of computer software to better help plan, design, and operate transportation systems.

Graph Theory(GT) - Is the study of graphs, which are mathematical structures used to model pairwise relations between objects.

Machine Learning(ML) - Is the study of computer algorithms that can improve automatically through experience and by the use of data. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so.

operating system(OS) - Is system software that manages computer hardware, software resources, and provides common services for computer programs.

Smart city(SC) - Is a technologically modern urban area that uses different types of electronic methods, voice activation methods and sensors to collect specific data.

Traffic Laws(TL) - Traffic laws are the laws which govern traffic and regulate vehicles.

Traffic Flow(TF) - Traffic flow is the study of interactions between travelers (including pedestrians, cyclists, drivers, and their vehicles) and infrastructure (including highways, signage, and traffic control devices), with the aim of understanding and developing an optimal transport network with efficient movement of traffic and minimal traffic congestion problems.

road infrastructure(RI) - It consists of the installation of fixed assets including surface roads and railways and terminals such as bus stops, trucking terminals, railways stations.



3 INTRODUCTION

The general field of our project is Smart Cities(SC). The rapid growth of human population, spoilage of natural resources, increasing population and number of vehicles in the urban life around the world are in a desperate need of a modern solution which the existing model of a city has we know it cannot provide. To solve these issues a new city model has been proposed in past several years known as a Smart City [1]. Also called eco-city or sustainable city, the smart city aims to improve the quality of urban services or reduce its costs. In the past few years many governments and cities around the world adopted this approach and implemented it in many different ways. Implementations and actions already taken in the field of smart cities are adequate water supply, assured electricity supply, sanitation, including solid waste management, efficient urban mobility and public transport, affordable housing, robust IT connectivity and digitalization, good governance, sustainable environment, safety and security of citizens and more. Examples for such cities are Singapore, Barcelona, San Francisco, London and more. Building a smart city or transforming a city into one is no easy task, and there are many challenges to take into account. Infrastructure and costs, security and privacy concerns and social risks are a partial list of the challenges in building a smart city. Despite these challenges a smart city has many practical, economic benefits such as environment impact, optimization of energy and water management, transportation, security and many more.

3.1 DEFINING THE PROBLEM

In the past decade traffic congestions has become an inescapable condition in large and growing metropolitan areas across the world [2]. Traffic congestions are getting worst with the growth of human population and increasing ratio of human-vehicle. The current traffic architecture and systems cannot handle the number of vehicles on the road and a new solution for inner city transportation is required.

3.2 THE TECHNOLOGICAL CHALLENGE

There are couple of reasons that a system such as DTLS has not yet been invented. The DTLS is a futuristic system that cannot be implemented without some core futuristic dependencies such as all vehicles will be fully automated with no exceptions, wide spread of 5G networks all around the city, social and government conception and more. No such city exists yet and, in this project, we will simulate such futuristic city. The growth of AV industry, 5G networks, cloud computing and efficient hardware technologies enables the DTLS possible existence. Assuming the DTLS exists, up and running in a smart city, transport congestions will be nearly eliminated within this city. The reason is that DTLS can make such a difference is a new innovative way of thinking about road infrastructures. Imagine a major highway leading in and out of the city on a Monday morning, one side of the road for entering the city is highly congested and the other way is almost completely empty as shown in **Error! Reference source not found.** It is clear that this is not an optimized way of using road infrastructures. In such situation the DTLS will open both ways in the direction of entering the city and use alternative ways for leaving the city.



FIGURE 1: ONE WAY TRAFFIC CONGESTION



4 WAYS TO SOLVE THE PROBLEM

Solution 1 (Preferred):

Our primary objective is to solve the problem by creating a traffic simulator(TS) that will simulate dynamic traffic laws in representative area such as Manhattan New-York. The implementation is based on an existing TS which we will extend and change according to our needs. Such TS can be found in the article: [3]. The core data structure of the code will be based on Graph Theory(GT) [4] where every node will be represented by a vertex and every road will be represented by an edge in the graph. The TS will have two main functionality modes:

1. TS under static traffic laws.
When using static traffic laws TS, the simulation will simulate the current way that traffic infrastructure is used as for today.
2. TS under dynamic traffic laws.
When using dynamic traffic laws TS, the simulation will simulate the result of the implementation of DTLS.

When executing the main function both modes will run in parallel. The simulation will show both runs side by side while collecting and presenting throughput and data about the simulation. The program architecture will be based on containerized environment. Each TS will be in his own container, another container with a database and one more container with Grafana. All the containers will share the same network and communicate with each other.

Solution 2:

This solution consists of two major parts:

1. Two AVs:
 - 1.1. Hardware: The vehicles are equipped with Arduino-based self-driving systems, as well as Jetson-based case and response systems. The Arduino will control driving related hardware such as motors. And the Jetson will control all the machine learning(ML) related hardware such as camera, Lidar, etc.
 - 1.2. Software: The Arduino micro-controller will be loaded with routine software to control all the hardware related to driving. This software will be written in C language using the Arduino IDE. The Jetson Nano will be loaded with Ubuntu OS which will run the ML related software using Python language and additional ML frameworks such as TensorFlow. In addition, the Jetson Nano will hold the current map and traffic laws and feed them to the Arduino. The Jetson will communicate with the main server using WIFI sending data about congestions and obstacles and receiving updated map and traffic laws from the server. whenever a new map or laws update is received by the Jetson it will pass it on to the Arduino.
2. DTLS(Dynamic transport law system) software and infrastructure:
 - 2.1. Hardware: The main server will be a personal computer with Ubuntu OS. The main server will not require special computing power for this project and thus it can be any personal use computer. The minimal requirements from the server are having WIFI communication abilities.
 - 2.2. Software: The DTLS will run on the main server. Receiving data from the vehicles analyzing it and making map and laws decision and sending those decisions in the form of a new updated map and a set of traffic laws. All data transformation will be passed using JSON format.

Our approach for implementing the transportation in a smart city is innovative and futuristic. A smart city transportation using AV provides us with new solutions and opportunities [5]



5 EXPECTED PROJECT ARTIFACT

In the end, we expect to show the ability of creating dynamic traffic law system in a smart city by presetting a number of real-life situations where the DTLS can be evolutionary. We will present three screens, two of them will present the TS in real time. One screen with TS without dynamic traffic laws, and the other one with dynamic traffic laws enable. Both simulations will use the same probability inputs to preserve the data integrity. Each screen will present different behavior although the inputs are the same. The third screen will present a Grafana dashboard that was enriched with data from the simulations. The dashboard will present the throughput and more matrices differences between the simulations.

6 DESCRIBING A SIMILAR IDEA THAT CAN BE INSPIRING

Every Monday morning while driving to work in Tel-Aviv, we have a lot of free time for thinking while almost standing still in traffic jams on Ayalon high-way. Every time we look to the other side of the road leading outwards from the city, an uneasy feeling is passed through our body when seeing it almost completely empty. One morning we understood why. Seeing such inefficient and waste of road infrastructure, suddenly didn't make sense. These thoughts with the combination of prior knowledge of the concept of smart cities led us to this idea.

7 RISKS, UNCERTAINTY AND PROJECT CONSTRAINTS

This project requires integration of many different technologies. As a result there are many risks involved as follows:

1. Integration of different technologies:

Risk: The project includes many different technologies such as Python scripting language, Docker, Docker Compose, Grafana, Database and more. Thus, the task of integrating all the different technologies in an efficient and coherent manner requires a big amount of effort.

Possible solution: The current architecture can be changed at any given time. All of the source code is written with source control technology(Git), this allows us to build the project in many different branches in parallel without having one solution interfering with the other.

2. Innovative approach:

Risk: We present a new and innovative way of optimizing the usage of road infrastructure in a manner that has never been suggested before. This innovative project presents us with many unpredictable issues and unexpected obstacle as we go.

Possible Solution: Both of us are currently working in development teams in an agile environment. In this project we work according to agile software development methodologies, this allows us to react fast to changes in a dynamic project like this. Furthermore, we rely on many different aspects that we learned during our studies such as Data-structures, Algorithms, Software Engineering, Physics, Calculus and all the different coding language we have learned.

3. Constrains:

Fixed map area for the simulation: In this project our main goal is to show the correctness of DTLS. Thus, showing the correctness in a representing area such as Manhattan New-York satisfies our needs. The reason we do not allow a dynamic choice of map area is only because of the big effort it requires without serving our goal directly. Note – We build our code in a way that it is extendable and such feature could be merged in future version of the projects.

8 SIMILAR WORKS IN THE FIELD

1. Simulating Traffic Flow in Python: This practical article presents an efficient and relatively simple TS written in Python scripting language. Alongside all the theoretical aspects of developing a TS, the author published the entire source code as an open-source Git repository. Thus, this project is a great starting point for us to extend from. [3] The main reason behind simulating traffic is generating data without the real world. Instead of testing new ideas on how to manage traffic systems in the real world or collect data using sensors, you can use a model run on software to predict traffic flow. There are three main possibilities When implementing a TS:
 - Microscopic models - represent every vehicle separately and attempt to replicate driver behavior.
 - Macroscopic models: describe the movement of vehicles as a whole in terms of traffic density (vehicle per km) and traffic flow (vehicles per minute). They are usually analogous to fluid flow.
 - Mesoscopic models: are hybrid models that combine the features of both microscopic and macroscopic models; They model flow as "packets" of vehicles.

In our project as well as in this TS, the Microscopic model was chosen

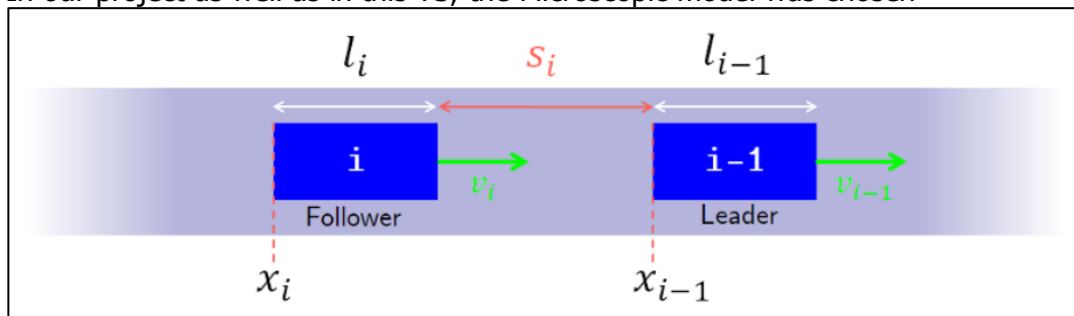


FIGURE 2: THE MICROSCOPIC TS MODEL APPROACH

. In the figure above we present how the basic concept of the model can be implemented when:

x_i – The i 'th vehicle.

l_i – The length of the i 'th vehicle.

v_i – The velocity of the i 'th vehicle.

s_i – The distance between the i and the $i - 1$ vehicle.

And from the above we can drive:

$$s_i = x_i - x_{i-1} - l_i, \Delta v = |v_i - v_{i-1}|$$

2. The Development of the Smart Cities in the Connected and Autonomous Vehicles (CAVs) Era: From Mobility Patterns to Scaling in Cities: A smart city transportation network needs to be clean and efficient. The main goal is Reducing traffic congestion inside the city as well as Reducing the CO2 footprint. In order to provide a city with Intelligent



transportation there is a need of many sensors and traffic management centers. Technologies that are currently being developed in the area of AV are crucial part of transforming a city into a Smart City [6].

3. Understanding autonomous vehicles: One of the key aspects of A smart city is transportation. Due to investment in and development of road infrastructure and vehicle technology, AV are expected to be the next big breakthrough. In the future, drivers will not be required to drive their vehicles, but instead the vehicle will be supplied with the destination's address, and the vehicles will begin to travel the required route using traffic laws visible on the road and connectivity to the Internet [7].

9 FUNCTIONAL REQUIREMENTS

9.1 PROJECT REQUIREMENTS

The system shall run the simulations according to the generated inputs. The system should be able to calculate and analyze the data in real time as well as deep data analysis after execution. The system should be able to handle parallel executions up to 60 frames per second.

9.2 PROJECT OPERATION AND USAGE

The programs main window shall have three main functionalities as shown in figure 3.

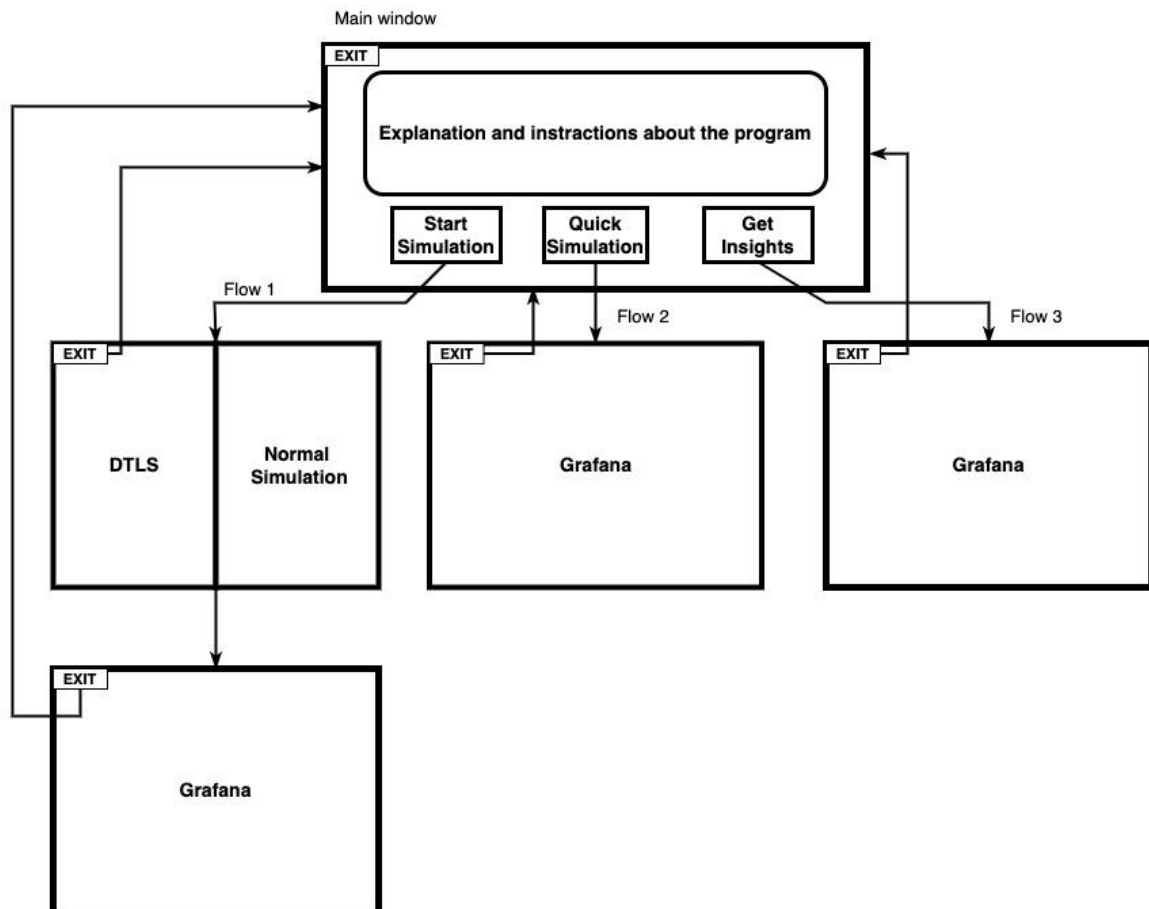


FIGURE 3 : SYSTEM IMPLEMENTATION



The main window will present a short explanation about the projects goal and implementation as well as short instructions regarding how to use the program. A quick link to the programs documentation will be added as well. Below, there will be three buttons:

1. Start simulation(flow 1 in figure 3): When clicking on "Start simulation" the user will be redirected to a new window split to two. Both sides shall present a simulation in real time while showing dynamic metrics. One side with DTLS, and the other side with normal TS. After both simulations end, a Grafana window will be presented with insights about the simulations.
2. Quick simulation(flow 2 in figure 3): When clicking on "Quick simulation" a quick simulation will start in the background. This option is used for getting quick insights in short time. After the background simulation finishes a Grafana window will be presented with insights about the simulations.
3. Get insights(flow 3 in figure 3):): When clicking on "Get insights" a interactive Grafana dashboard will be presented. The dashboard will have the ability to present data with queries from past simulations, Thus the dashboard will be able to present data from a single simulation or from a combination of many simulations, or even all time.

9.3 SYSTEM REQUIREMENTS

The program inherits the Pygame python library and thus it also inherits the library minimum system requirements which are:

- OS: 64-bit Windows 7 or later or OS X 10.11 or later.
- Processor: 1.5GHz or faster.
- Memory: 4GB (4,096MB) RAM.
- Free HDD space: 3GB.
- Monitor, mouse and keyboard.

Software requirements:

- Docker.

10 WHAT HAS BEEN DONE SO FAR IN THE PROJECT

At the beginning, the project was to build a simple AV based on Jetson Nano with the following goals: detecting objects, autonomous driving to target point, detecting pedestrians. After the two meetings with our instructor we felt the project is not sophisticated or interesting enough in our opinion. So, we asked the collage management for the permission to redefine the project to a new one which is more complicated and challenging. After a short while, we received the green light and presented our new idea for the project. The new idea was to create a system with AVs where the traffic laws changed dynamically and the AVs act accordingly to the dynamic set of laws.

After the redefinition of the project we started working on possible implementations. The selected implementation was using two AVs each one is comprised of a Jetson Nano, Arduino, Lidar sensors and a camera. Alongside the AVs there will be a main server with all the software related to managing the dynamic traffic and sending the data back and forth to the AVs. The first task we had is learning about the Jetson



Nano and ML. As for ML, we took a practical course led by our instructor. To learn about the Jetson Nano, we took a Jetson Nano from the collage logistic department and took an online course for learning about it. As part of the course, we needed to install an operating system(OS) and we encountered an issue with burning the OS on the memory card we received. After reading the minimum requirements we saw the memory card wasn't subtle and we asked for a new one from the logistic department. After the card replacement we managed to burn the OS on the card and started working with the Jetson Nano.

Following the work we have done on the Jetson Nano, we got to the understanding that the implementation we have choose does not serve the goal we are trying to accomplish in a complete manner. The reason is that using only two AVs, is very limiting and does not produce enough data in order to present the correctness of the basic assumption we have made. Which is using the DTLS leads to maximum utilization of the road infrastructure. After these insights, with the collaboration of our instructor, we came up with a new and improved implementation which is converting the project to pure software solution which is a TS.

Moving to a software solution allows us to produce an exponentially larger sets of use cases, which represent the real life in much better precision. Furthermore, the DTLS is a complex system which includes hardware and software aspects. The hardware related to the core functionality of the DTLS is having fully automated AVs, which are integrated to work with the DTLS software interface. In this implementation we implement the software aspect of the system, without the effort of the hardware aspect. This enable us to focus on the logical aspects of the system and hence get optimal results according to the time invested in the project.

After the realization that we need a TS, we started thinking about different technologies and software languages, that we are going to use in order to build the simulator. It was clear to us from our research an prior knowledge that we are going to use Python scripting language. There are two main reasons why we choose Python, the first one is the rich set of different libraries such as gamming and GT. The second one is the simplicity of the language. At the starting point with the guidance of our instructor we got to realization that there are two ways to proceed. The first one is writing the simulator from scratch and the other one is finding an existing open-source simulator that we can extend to our needs. Following a long research we have found an open-source TS written in Python with full documentation which was a great starting point for our project. The documentation includes a detailed and straightforward article about the simulator [3]. We presented this simulator to our instructor and he approved. The chosen simulator uses Pygame library as the main framework for the simulation. The simulator dynamically creates the traffic infrastructure(roads, intersections, headlights, etc...) and populate it with vehicles according to probability parameters in the simulation.

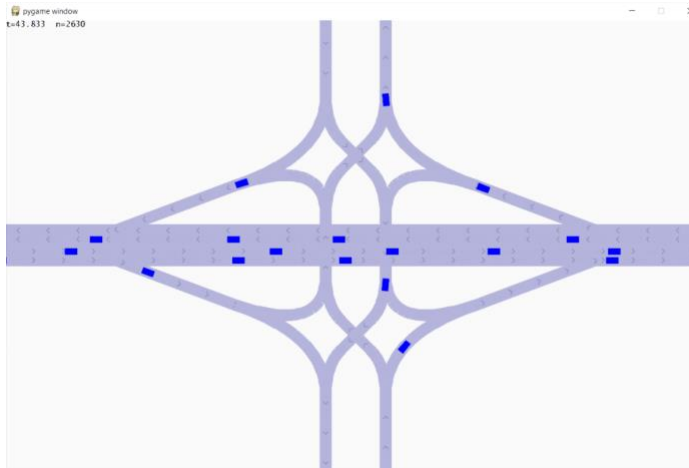


FIGURE 4: EXAMPLE OF THE BASE SIMULATOR

As shown in figure 4, in the simulator each road has a direction according to the arrows, and a vehicle is represented by a blue rectangle.

Initially we started from reading the article carefully and understanding all the theoretical aspects of the simulator. Only after we fully understood all the theoretical aspects, we started learning the code. The code is written in Object Oriented Programming (OOP) concept. The core classes of the program are:

- Window class – This class uses the Pygame library and controls all the UI aspects of the simulation.
- Vehicle class – This class represents a single vehicle in the simulation.
- Vehicle Generator class – This class generates vehicles using the vehicle class according to probability inputs.
- Road class – This class represents a single road in the simulation holding zero or more vehicles.
- Simulation class – This class initializes all the different classes and parameters in order to start a simulation and controls the data flows between them.

The existing simulator is not suitable for presenting real life. In order to improve the simulator we added the following features:

Distinct vehicle – The simulator supports only one kind of vehicle. We added the support of three different kinds of vehicles. The distinction between them is supported in two levels, the first is by different colors and sizes in the UI, and the second is the size and speed of the different vehicles as presented in the memory. The three vehicles representation are:

- Small – this representation is used for A' driver license category (as shown in figure 5) and the color we choose for the category is Red (as shown in figure 6).
- Medium – this representation is used for B' driver license category (as shown in figure 5) and the color we choose for the category is Green (as shown in figure 6).
- Large – this representation is used for C' and D' driver license category (as shown in figure 5) and the color we choose for the category is Blue (as shown in figure 6).

AM	C1	CE
A1	C	D1E
A2	D1	DE
A	D	L
B1	BE	T
B	C1E	

FIGURE 5: DRIVER LICENSE CATEGORIES

The way we implemented the different speeds for each category and also made each vehicle (even if from the same category) speed different is by defining the average speed and speed range boundaries for each category, and using a random number (can be positive or negative) give each vehicle a random speed value within their allowed range. Mathematically: given an allowed range R for category x

$$Speed_{car} = Speed_{average} + random,$$

$$Speed_{average} = \frac{speed_{max} + speed_{min}}{2}$$

$$0 \leq random \leq \frac{speed_{max} - speed_{min}}{2}$$

$$speed_{max} - speed_{min} \in R$$

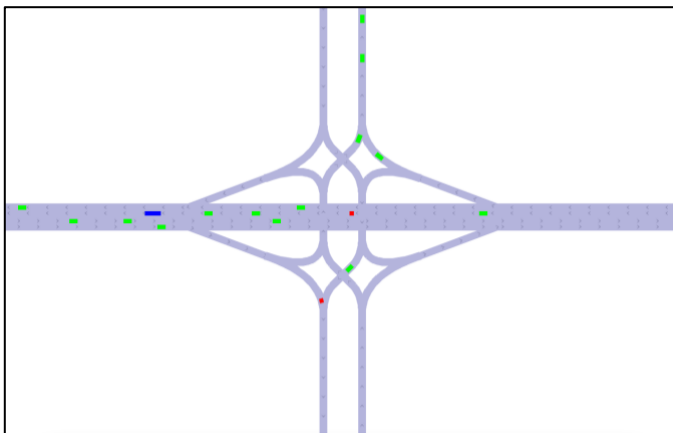


FIGURE 6: DISTINCT VEHICLES

As part of data collection we started presenting metrics about the simulation. The metrics can be found on the top left corner of the simulation window (as shown in figure 7). As for now, only two metrics are shown. The first one is the total number of vehicles in the simulation at any given time, and the second metric is the time passed in seconds from the execution time. Many more metrics will be added in the future.

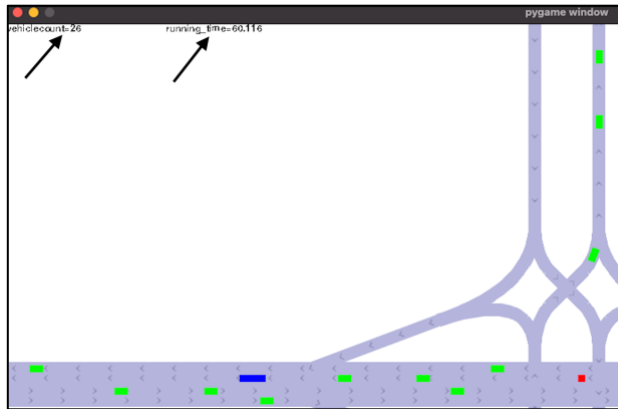


FIGURE 7: SIMULATION METRICES

The next challenge we encountered was creating new simulations including roads, intersections, headlights, etc.. Creating new simulations with the current implementation is exhausting, because of none generic and poor architecture in the simulator. The creation of each element is specified with fixed pixels and that makes building large and complex simulations to nearly impossible. The solution we came up with is adding a customized background photo with X and Y axis to map each pixel in the simulation(as shown in figure 8).

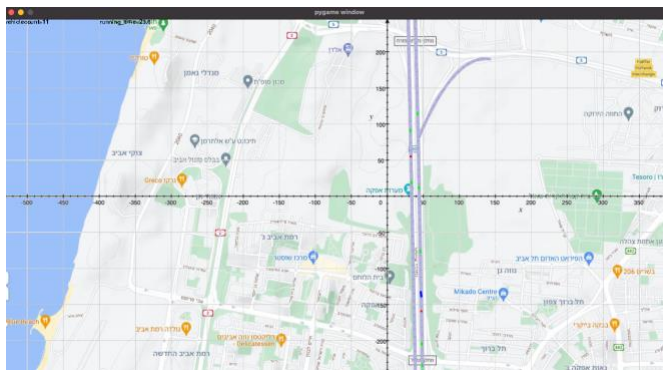


FIGURE 8: PIXELIZED MAP

The new feature we are currently working on is modelling the map as a graph. Each intersection will be represented as a vertex and each road as edge in the graph. Modeling the map as a graph allows us to use GT algorithms and hence get better performance and resilience. This feature has a direct impact on all of the programs modules and that makes it a large and complex task. Furthermore, using a graph allows us to make quick changes in a generic way on the map in real time. This is very important for the next and most major phase of the development which is implementing the DTLS from the TS.



11 PLANNING

MILESTONES TABLE 11.1

Milestone number	Milestone description	End date	Total man hours	Artifact
1	Preparing report	24/10/2021	25	Preparing report
2	Simulation validation	4/1/2022	140	Simulation running
3	Progress report	16/1/2022	50	Progress report
4	GT and DTLS integration	3/7/2022	300	Simulation running with GT under DTLS
5	Projects day + practical presentation	12/7/2022	55	Poster + presentation + POC
6	Data visualization	8/9/2022	120	Present all the collected data in Grafana
7	Defenses	15/9/2022	40	Project book + working project

TABLE 1: MILESTONE TABLE

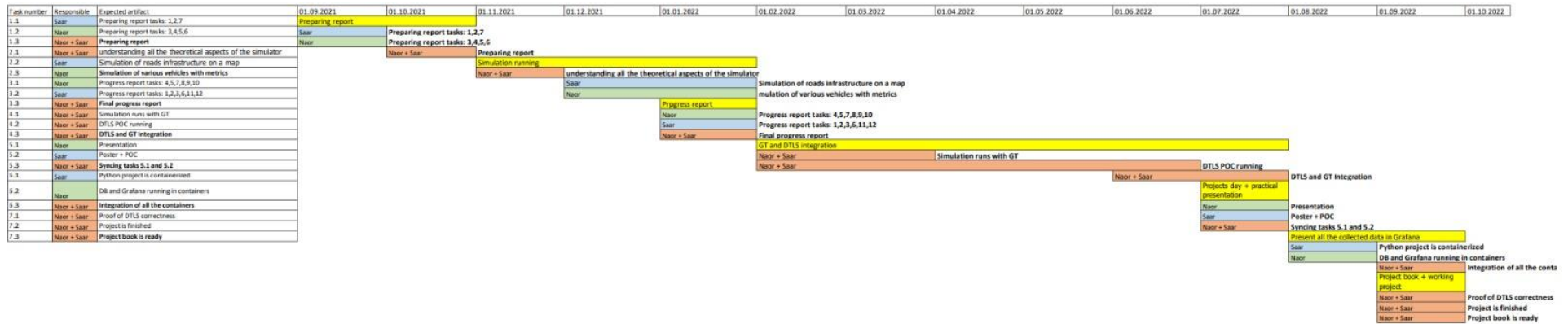


11.2 TASKS TABLE

Tasks Table							
Milestone number	Milestone description	Task number	Start date	End date	Total working hours	Responsible	Expected artifact
1	Preparing report	1.1	15.10.2021	23.10.2021	10	Saar	Preparing report tasks: 1,2,7
		1.2	15.10.2021	23.10.2021	10	Naor	Preparing report tasks: 3,4,5,6
		1.3	23.10.2021	24.10.2021	25	Naor + Saar	Preparing report
2	simulator validation	2.1	25.10.2021	8.11.2021	40	Naor + Saar	understanding all the theoretical aspects of the simulator
		2.2	09.11.2021	25.12.2021	50	Saar	Simulation of roads infrastructure on a map
		2.3	09.11.2021	25.12.2021	50	Naor	Simulation of various vehicles with metrics
3	Progress report	3.1	26.12.2021	10.01.2022	20	Naor	Progress report tasks: 4,5,7,8,9,10
		3.2	26.12.2021	10.01.2022	20	Saar	Progress report tasks: 1,2,3,6,11,12
		3.3	11.01.2022	15.01.2022	25	Naor + Saar	Final progress report
4	GT and DTLS integration	4.1	16.01.2022	15.03.2022	120	Naor + Saar	Simulation runs with GT
		4.2	16.01.2022	10.06.2022	50	Naor + Saar	DTLS POC running
		4.3	11.06.2022	3.7.2022	130	Naor + Saar	DTLS and GT Integration
5	Projects day + practical presentation	5.1	04.07.2022	09.07.2022	15	Naor	Presentation
		5.2	04.07.2022	09.07.2022	15	Saar	Poster + POC
		5.3	10.07.2022	12.07.2022	25	Naor + Saar	Syncing tasks 5.1 and 5.2
6	Data visualization	6.1	13.07.2022	20.08.2022	40	Saar	Python project is containerized
		6.2	13.07.2022	20.08.2022	40	Naor	DB and Grafana running in containers
		6.3	21.08.2022	08.09.2022	40	Naor + Saar	Integration of all the containers
7	Defenses	7.1	09.09.2022	11.09.2022	15	Naor + Saar	Proof of DTLS correctness
		7.2	12.09.2022	12.09.2022	5	Naor + Saar	Project is finished
		7.3	12.09.2022	15.09.2022	20	Naor + Saar	Project book is ready

TABLE 2: TASKS TABLE

11.3 GANTT



12 REFERENCES

- [1] N. Radovan, K. Radek and K. Jaroslav, "Smart City Concept, Applications and Services," *Journal of Telecommunications*, vol. 3, p. 8, 2014.
- [2] A. Downs, "brookings," 1 January 2004. [Online]. Available: <https://www.brookings.edu/research/traffic-why-its-getting-worse-what-government-can-do/>. [Accessed January 2004].
- [3] Available: .[מקור]. B. Himite, "Simulating Traffic Flow in Python," 5 September 2021 .<https://towardsdatascience.com/simulating-traffic-flow-in-python-ee1eab4dd20f>
- [4] P. Vasseur, "Spatial Modeling of Urban 1 O. Kamaldeep Singh, G. del Mondo, Y. Dupuis .pp. 1-15, 2017 ,*SAGEO* ", Road Traffic Using Graph Theory
- [5] S. Asadi Bagloee, M. Tavana, M. Asadi and O. Tracey, "Autonomous vehicles: challenges, opportunities, and future implications for transportation policies," *Journal of Modern Transportation*, vol. 24, pp. 284-303, 2016.
- [6] T. Campisi, A. Severino, M. Ahmad Al-Rashid and G. Pau, "The Development of the Smart Cities in the Connected and Autonomous Vehicles (CAVs) Era," *Infrastructures*, vol. 6, p. 21, 2021.
- [7] A. Faisal, T. Yigitcanlar, M. Kamruzzaman and G. Currie, "Understanding autonomous vehicles," *Journal of Transport and Land Use*, vol. 12, pp. 45-72, 2019.
- [8] B. Macy , "techrepublic," 5 February 2020. [Online]. Available: <https://www.techrepublic.com/article/autonomous-vehicles-how-7-countries-are-handling-the-regulatory-landscape/>. [Accessed 5 February 2020].



APPENDICE

APPENDICES LIST

Appendix A – Block diagram



APPENDIX A – BLOCK DIAGRAM

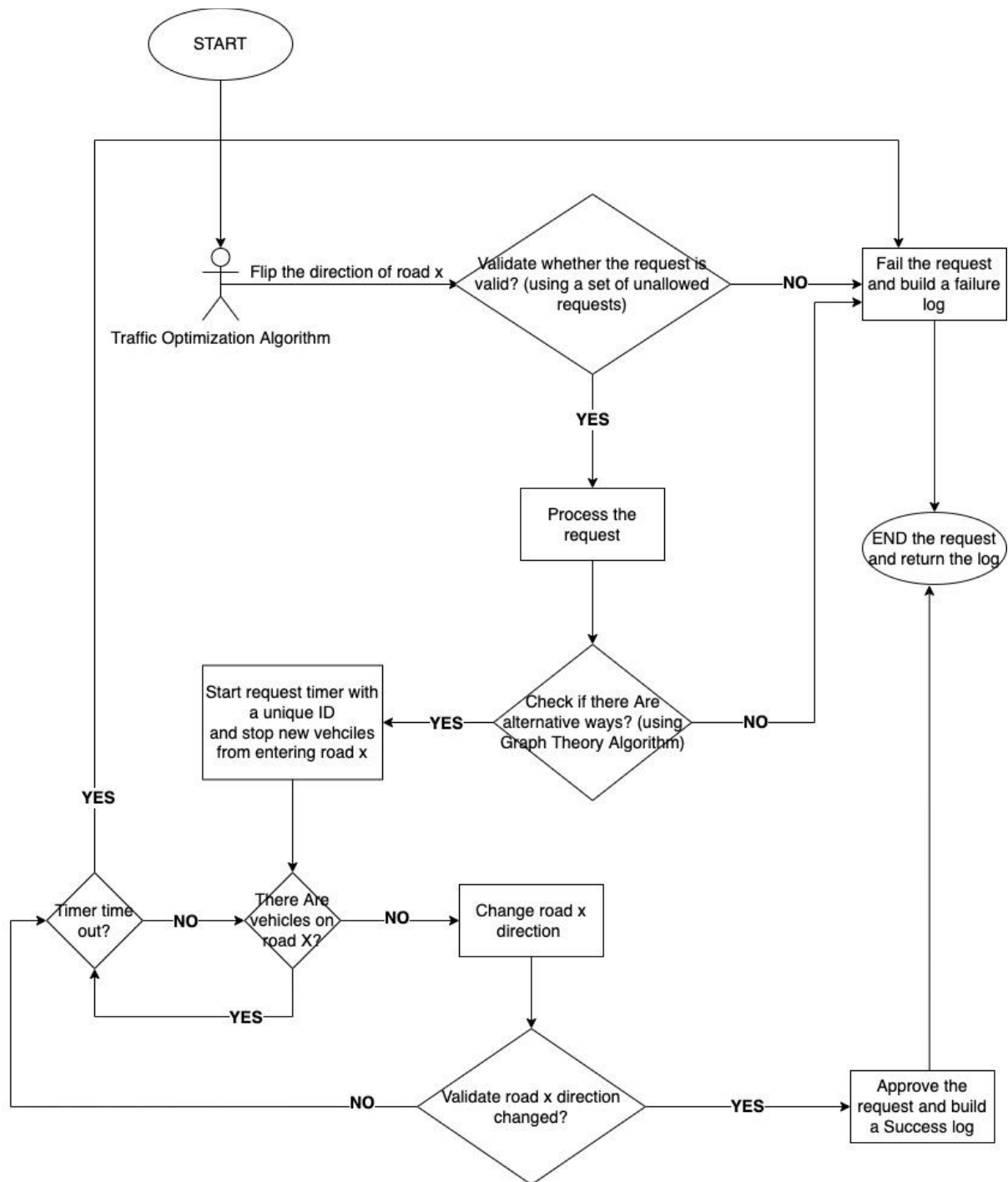


FIGURE 9 : BLOCK DIAGRAM