POLITECNICO DI TORINO

Collegio di Ingegneria Meccanica, Aerospaziale, dell'Autoveicolo e della Produzione

Master of Science degree in Automotive Engineering

Master Thesis

Digital Twin at Urban Scale for a Master Plan in the Area of the CEBU Airport (Philippine)

Supervisors

Prof. Anna Osello
Prof. Francesca Maria Ugliotti

Candidate

Zhang Zheyuan

May.2021

Acknowledgement

I want to express my sincere gratitude to our supervisors prof. Anna Osello and prof. Francesca Maria Ugliotti for let me join this stimulate case study project, for their valuable advice and guidance during the work, and for their always supportive on my research interest, also on the resources both for software and hardware. Their encouragement motivates me to keep studying and complete this project, let me enjoy the working process.

Thanks to my group mate civil engineering student Daniele Iunti, when I encounter difficulties, he always helps me at any time and his many excellent ideals improve the project qualities.

Thanks to another group mate and also my girlfriend Ma Fanshu, she is always patient and positive not only during the work process of this project, but also during the whole master study courses. Thanks to her accompany and intelligence, we can learn from each other and progress together.

Thanks to my mom for raising me up and always support my study and hobbies.

Abstract

As the fast development and urbanization of Metro Cebu in recent years, people density has steeply increased, the conflict between the traffic demand and the insufficient road network infrastructure becomes more and more serious. How to alleviate the traffic congestion and pollution problems with the help of new generation technologies such as digital twin, IOT, artificial intelligence, virtual reality etc. becomes an interest topic.

The case study aims to investigate the methodological approach behind the implementation of a Digital Twin, system supposed to supply a smart management, as a part of a master plan focused on improving the urban mobility of a limited portion of Mactan Island, Philippines. As a part of the project, this thesis aims to describe the actual traffic situation of Mactan island, showing a possible implementation addressing the urban mobility problem, build the traffic environment of interested area in the digital world and make the VR technology-based driving simulator in the traffic environment. the methodological Digital Twin supplied aims to enhance citizens life quality with the help of industry 4.0 technologies. This objective is pursued addressing some of the seventeen sustainable development goals by the UN 2030 Agenda.

Table of Contents

Contents

1. Introduction

As the explosive increasing of population in recent decades, urban, as a densely populated area, has an increasing demand for transportation, the increase of number of vehicles has brought great challenges to the city management and environment. How to use the emerging technologies such as electric mobilities, renewable energy, internet of things, artificial intelligence to build a convenient and fast, easy-to-manage and sustainable transportation system has become a hot topic. In this chapter (**1. Introduction**), an overview of traffic pollution and its relative factor will be presented (**1.1 The impact of traffic on the environment**), the population, traffic and energy structure state of Metro Cebu will be discussed (**1.2 The development status of Metro Cebu**) and the target of this case study on Cebu Mactan airport area will be mentioned (**1.3 Target of Case Study and Structure of Thesis**).

1.1 The impacts of traffic on the environment

In the 21st century, with the increase in the number of cars and the wider range of use, automobile exhaust pollution has become a global problem, automobile emit many substances which bring the negative effects, especially those endanger the urban environment and causing the respiratory diseases of human, the citizens have become the direct victims of automobile exhaust pollution.
In various countries of the world, automobile exhaust pollution is no longer a new topic, since 1940s, there have been more photochemical smog incidents in cities such as Los Angeles in the United States and Tokyo in Japan, this incident caused many casualties and huge economic losses, we can say that cars are a mobile source of pollution. [6]
There are about 7 million people die every year due to air pollution from the study of the world health organization (WHO). In Southeast Asia, most of this air pollution comes from traffic in highly urbanized areas.

1.1.1 Main emissions of traditional engine vehicles

Traditional engine vehicles exhaust pollutants mainly include Carbon Monoxide (CO), Hydrocarbons Compound (HC), Nitrogen Oxide Compound (NOx), Sulfur Dioxide (SO2), Particulate Matter (PM) and Greenhouse gases (ex. CO2).
Carbon monoxide (CO) is an intermediate product of hydrocarbon fuel combustion, it has toxic effect on human and animals, be generated mainly due to the incomplete combustion of the fuel caused by insufficient oxygen in the combustion chamber of the engine or low temperature

conditions, when the car is overloaded, driving slowly or when running in neutral gear, the fuel cannot be fully burned, and the carbon monoxide content in the exhaust gas will increase.

The hydrocarbons in car exhaust come from 3 emission sources, for gasoline engines, about 60% of the hydrocarbons come from internal combustion engine's waste, 20% - 25% come from the leakage of the crankcase (PCV system), the remaining 15% - 20% comes from the evaporation of the fuel system. Some hydrocarbons can participate in the creation of greenhouse effect.

Nitrogen Oxide (NOx) is a kind of brown smelly gas, it is harmful for the human respiratory system, and it can participate in the formation of acid rains and smog. The main reason to produce nitrogen oxides is the high temperature and oxygen-rich environment. From the perspective of the combustion process, more than 95% of the emitted nitrogen oxidizes is nitrogen monoxide (NO) and the rest is nitrogen dioxide (NO2). [6][7]

Particulate matter (PM) is a generic term for all particles emitted in the air, it has many sources such as soot in the exhaust gas, wearing of the brake and tire etc. it is a very fine particles (less than 1/10 of human hair) which very harmful for the human's health as it can penetrate deep into the lungs.

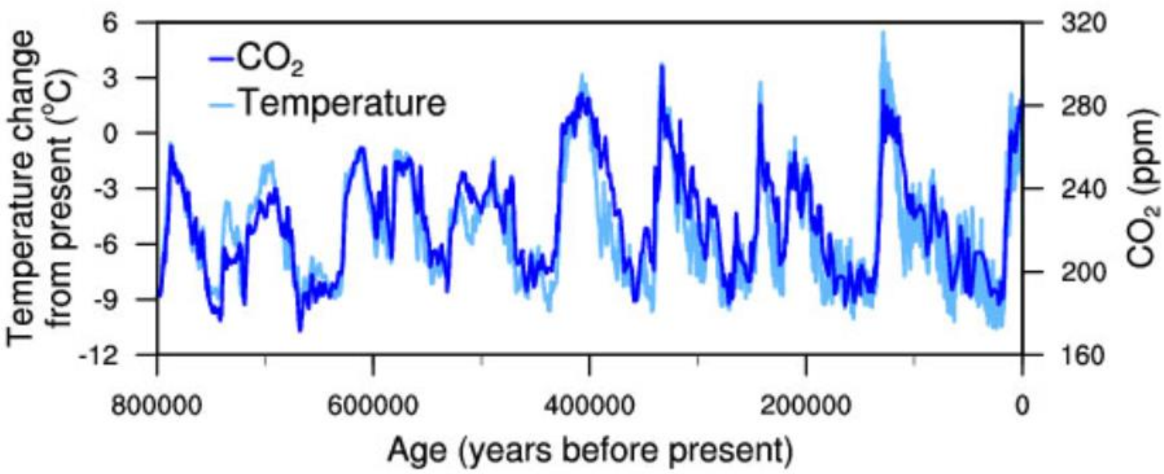


Figure 1.1 CO2 and Temperature [8]

The carbon dioxide (CO2) does not hurt the people's health, but as the heat-trapping gas, CO2 contribute to the global climate change, as shown in **Fig. 1.1**, CO2 has a strong relationship with the temperature.

1.1.2 Emission relative to electrical vehicle

An electric car is a new type of car powered by a battery and driven by the electric motor, the first electric car was born in 1881, that was early, but due to the limitations of the battery technology, the development of electric car was very slowly, until after the 1970s, with the development of the battery technology, the car companies have started to develop electric car.

In recent years, the countries in the world continuously increase the effort to the environment protection and emission reduction, the electric car, as a transportation tool with low noise, low emissions, and good comprehensive utilization of energy, has been fast developed, and been seen as an important way to alleviate the energy crisis and environmental pollution in the new century.

As mentioned in the section **1.1.1,** traditional cars produce a lot of harmful gas during the operation, it not only gives pressure on environmental governance, but also harms human health, electric vehicle are driven by batteries, there is almost no exhaust emissions during operation, the direct exhaust emissions are reduced by more than 90% compared to traditional fuel vehicles, but electric vehicle consumes electricity instead of fuel, and in the most of countries, electricity is still mainly generated from the conventional fossil fuels, therefore, the result increase in exhaust emission from power plants should be included in the emission of electric vehicles. [9][10]

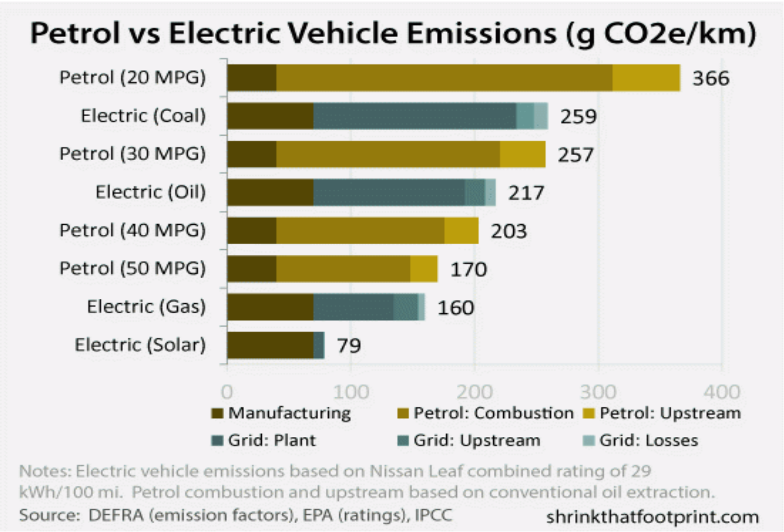


Figure 1.2 Petrol vs Electric vehicle emission

As shown in the **Fig. 1.2**, the emission of electric vehicle is highly dependent on its energy sources, if the electricity is come from the conventional fossil fuels (Coal and Oil), the overall emission of electric vehicle can be higher than the petrol car with high fuel economy, if all electricity of electric vehicle is provided by the renewable energy source such as solar, hydro, wind etc., it can be seen as zero emission during operation phase.

On the other hand, compared with the power unit of the conventional vehicle, the battery life is significantly shorter, therefore, the emission relative to manufacturing and recycling of the batteries should be considered.

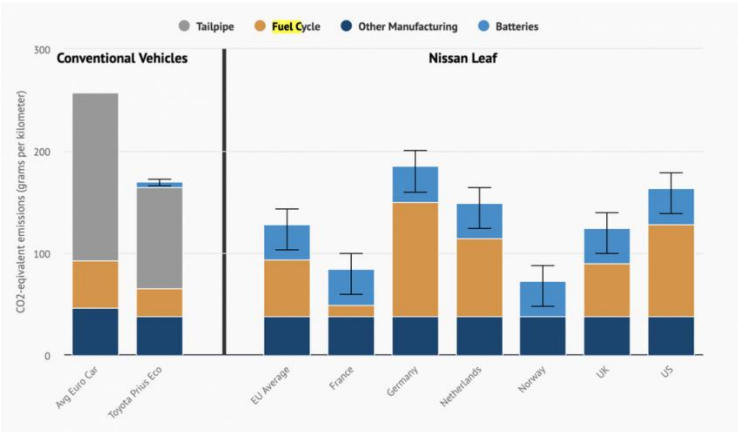


Figure 1.3 Electric and conventional vehicle life cycle emission [11]

As shown in **Fig. 1.3**, the manufacturing emission of electric car is higher than the conventional vehicle due to the battery production.

In summation, improving the structure of electric energy and making it develop towards a cleaner power generation portfolio is fundamental to reducing the CO2 emissions of electric vehicles, at the same time, improving the battery performance of electric vehicles can effectively reduce the impacts on greenhouse gases due to the production and recycling of the electric vehicle's batteries.

1.1.3 Emission and consumption relative to the vehicle speed

Because of the rapid increase in the number of motor vehicles in recent years, traffic congestion and environmental pollution have attracted more and more attention, the traffic congestion increases the idling time of motor vehicles, causing the vehicle stop and start frequently and decreases the vehicle average speed in the urban area, the influences on the emission for the traditional internal combustion engine car and influences on the energy consumption for the electric vehicle will be discussed in this section.

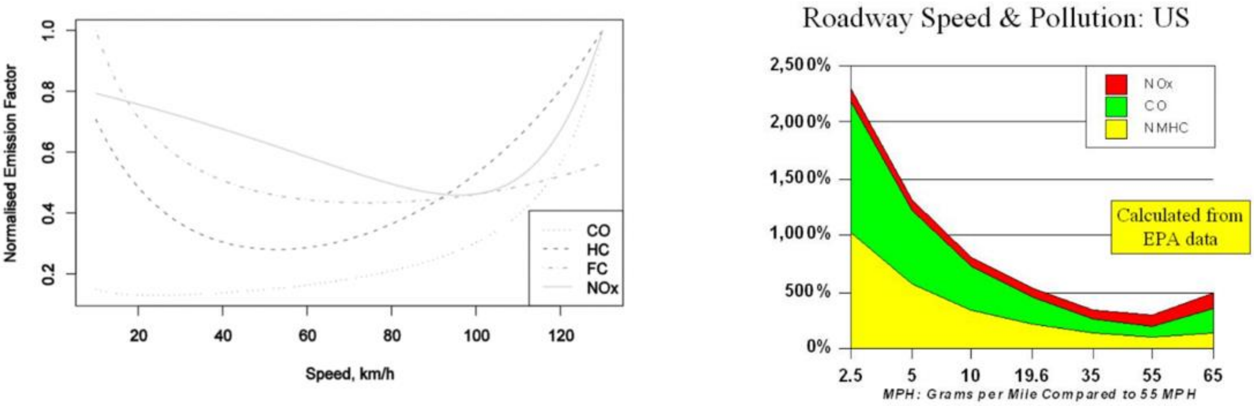


Figure 1.4 Emission VS Speed [12][14]

A large number of research results in different country based on different model showing that vehicle speed has an important impact on vehicle exhaust emissions, when the vehicle idling on the road, the engine is not operating at its peak temperature which will causing the incomplete burning of the fuel and increase the emission. All the result showing that for traditional internal combustion engine vehicle, the traffic congestion has direct relationship with its emission. [12][13][14]

Considering the emission from the electric vehicle cannot be directly measured during the operational phase, but it is strongly relative to the structure of its electricity energy sources which is variable from country to country, for the electric vehicles, the energy consumption is generally discussed.

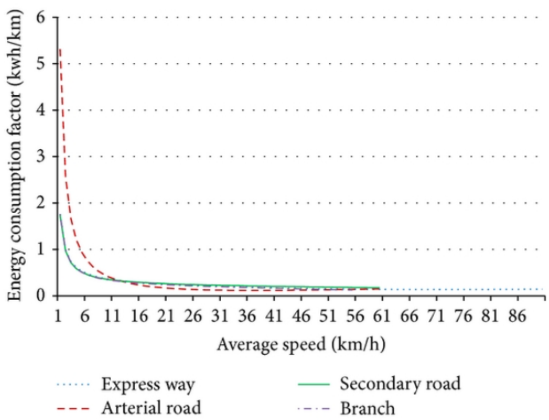


Figure 1.5 Energy consumption of electric vehicle under different road conditions and speed [15]

There are many real roads test-based research relative to the energy consumption of the electric vehicle. Result shows that the energy consumption is strongly relative to the traffic status, when traffic congestion happens, the speed of vehicles on the road are mainly distributed in the low area, since the start phase is more energy consuming, the energy consumption per ditance in the congested urban area is more than consumption in the non congested periferal area and on the highway. [15][16][17]

1.1.4 Section Summary

Traffic congestion will extend the travel time, affect the surrounding environment, damage the travel experience of travelers, and degrade the quality of life of urban residents. Under urban congestion conditions, the average vehicle speed is low, the vehicles accelerate and decelerate more frequently and the time ratio of vehicle under idling condition is high, due to the lower energy consumption under idling condition, the electric vehicle's overall efficiency and energy consumption is better than the traditional vehicle. Pure electric cars do not produce exhaust emissions by themselves, it is impact on the environment is mainly concentrated in upstream companies, especially the pollution due to the power generation plant, so the emission relative to the electric vehicle is highly depends on the energy source structure of the country.

1.2 The development status of Metro Cebu

The Metro Cebu (or Cebu metropolitan area) is the second largest metropolitan area in the Philippine (after Metro Manila), Metro Cebu is formed by Cebu city, Mandaue city and Lapu-Lapu city, with a population of 2.85 million (2015), is the economic, trade and financial center of southern Philippine.

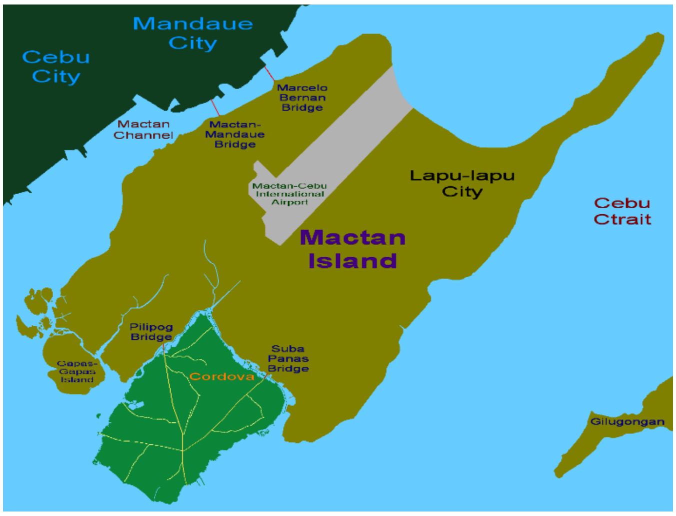


Figure 1.6 Mactan Island

The Lapu-Lapu city is positioned on the Mactan island and connect the main Cebu Island with two bridges (Sergio Osmeña Bridge and Marcelo Fernan Bridge). The Mactan-Cebu international airport is the second busiest airport in Philippine, it is located in the Lapu-Lapu city on the Mactan island.

1.2.1 Urbanization and population growth in Metro Cebu

With the development of social productivity, the progress of science and technology and the adjustment of the industrial structure, the society has gradually changed from a traditional rural society to a modern urban society. Positively, the increase of urbanization rate of the cities can create more job opportunities and absorb large number of surplus rural population, the labor force will gradually shift from the primary industry to the secondary, tertiary and fourth industry, but on the other hand, as the people gathering in the urban area, problems such as traffic congestion, resource shortage and the decline in the quality of life of urban residents are hindering the progress of the city.

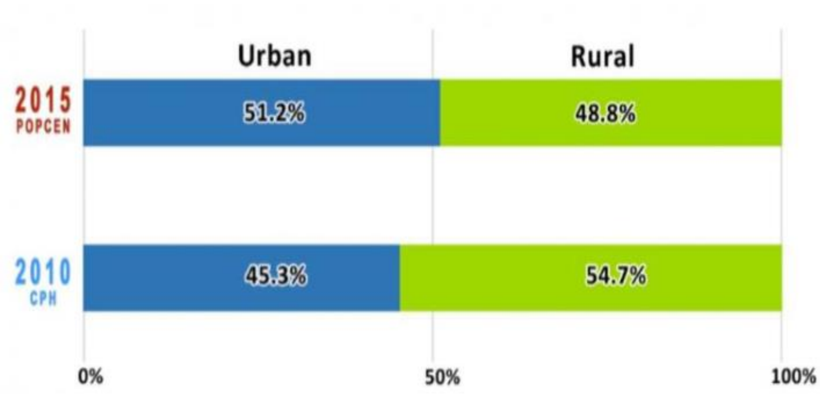


Figure 1.7 Urbanization rate of Philippine

In 2015, the level of urbanization or the percentage of population residing in urban areas in the Philippines was recorded at 51.2%, compare to the 45.3% in 2010, the potential and the increase speed of urbanization rate is significant. [1]

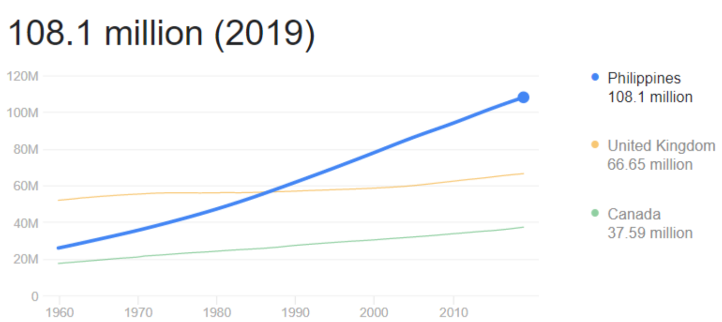


Figure 1.8 Population growth in Philippine

Meanwhile, the number of populations is growing rapidly in Philippine recent years, the increase of population promotes social consumption and stimulates economic growth, but it also brings challenges to the environment and urban transportation.

Under the effect of people growing and "Siphon Effect" of city, even if the urbanized area is enlarging, the people density will still increase steeply. In the list of population density by city, we can find Philippine has the greatest number of cities enter top 25, and it has 3 cities with the highest population density in the world. As the second largest metropolitan area in the Philippines, the population of metro Cebu is expected to increase from 2.85 million in 2015 to 5 million in 2050 due to rapid urbanization. [2][3]

1.2.2 The traffic status of Metro Cebu

Due to the increase of the traffic demand which caused by the population growth, economic development, the traffic condition in Metro Cebu is deteriorated rapidly, from the records of the Land Transportation Office (LTO) Regional Office VII, the number of registered private motor vehicle has been increased year by year, in which the majority registered vehicle type was motorcycle (more than 75%). Moreover, if we look at the **Fig. 1.9**, the percentage of motorcycle owned is the highest.

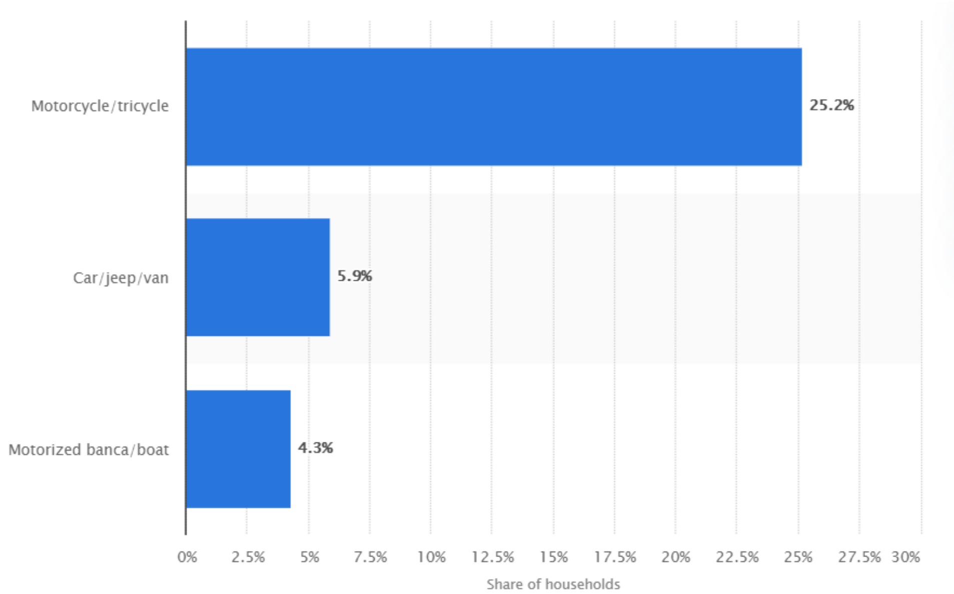


Figure 1.9 share of vehicle ownership among households in the Philippines in 2018

Road Classification	Length (km)
National Primary Road	121.0
National Secondary Road	191.0
National Tertiary Road	105.6
Provincial Road	112.5
City Road	171.1
Municipal Road	66.4
Barangay Road	630.6
Total	1,398.2

Source: JICA Study Team

Figure 1.10 Road Lengths in Metro Cebu by Road Classification

One reason is due to the relatively lower price of the motorcycle, another important reason is due to the road condition of Metro Cebu. Most of roads in the city are generally narrow with two lanes and 46% of roads are not paved (**Fig. 1.10**), the bad road condition is also an important factor which causes the traffic congestion, in this road and traffic condition, motorcycle as a flexible transportation tool, the advantage is obviously, it is easy to zip in and out from the crowed traffic and easy to park due to its small size relative to the car. [4]

Figure 1.11 Jeepney

For the public transportation in Metro Cebu, rail transit does not exist, the buses system is implementing but only has few terminals, Jeepney plays an important role, jeepneys were originally made from Jeep cars left by the U.S. Army after World War II. The term "Jeepney" comes from the combination of "Jeep" and "Jitney" ("shared taxi"), it is a small bus with franchise rights, which can flexibly arrange time to carry passengers on a fixed route. Jeepney are the most affordable tool

of transportation in Philippine, because of their open door design, it is convenient for both passengers and drivers to get on and off the bus, even if the jeepneys running at fixed route, but they can stop at any time, unlike buses that have fixed stops, this behavior brings convenience to passengers, but it also makes some jeepney's drivers become the source of traffic jam, they indiscriminately let passengers get on and off in the middle of the road, causing traffic to stop and endanger the safety of the passengers. Because of the frequently stop and go during the commuting, the engine of Jeepney is always work at low conversion efficiency, additionally, there are a large amount of Jeepney which is old and lack of maintenance, Jeepneys has become the main cause of air pollution of the city. To resolve the pollution problem of Jeepney, some city of Philippine has developed and implemented the e-Jeepney in some route, the e-Jeepney's passenger capacity is similar to the old one but driven by the battery and electric motor. [4][5]
The largest bus operator in Cebu province is "Ceres", it has a fleet of buses with a number around 300, these buses is used for the inter-urban transportation, half of them connect the Cebu city to the Bogo, where is positioned at the north of the province, another half fleet connect to the Santander, where located at the south of the province. For the bus commuting internal the Metro Cebu, "Mybus" is the only bus service which introduced in 2015 to primarily encourage the people to its mall, but in any way, the bus service plays a minor role in the public transportation system of Metro Cebu. [4]
In some area of the city, people have the demand of the public transportation, but due to the bad and narrow road condition, the public utility vehicles (PUV) such as Jeepneys, buses and taxis are hard to commute in such area.

Figure 1.12 "trisikads" (left) and "habal-habal" (right)

There are the tricycles "trisikads" and also a transportation method named "habal-habal" which is modified from the motorcycle and plays a function as motorcycle for hire or "motor taxi", this service is illegal but it can fulfill the requirement of people who has the public transport demand in these areas. As we can see from Fig. 1.6, these transportation methods are obtained mainly from the private modification of the motorcycle or bicycle, it is not safe for the passengers since the motorcycle has not been designed for the usage like this, considering the enlarging and pave the road needs long time, a cheap, flexible, and safe public mobility system for the people who need mobility service in these areas should be developed. [4]

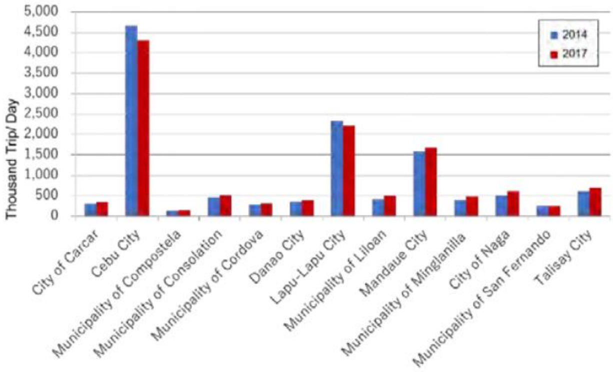


Figure 1.13 Trip generation and attraction between cities

Lapu-Lapu city, where the second busiest airport of Philippine (Mactan-Cebu international airport) located on, has attracted much traffic from other cities, and in the future, there will be more traffic considering the increasing trend from 2014 to 2017 (**Fig. 1.13**), that gives pressure to the two bridges, because they are the only two paths connect the Lapu-Lapu city to the main island of Cebu province.

Figure 1.14 Two bridges connecting Mactan island and main Cebu Island

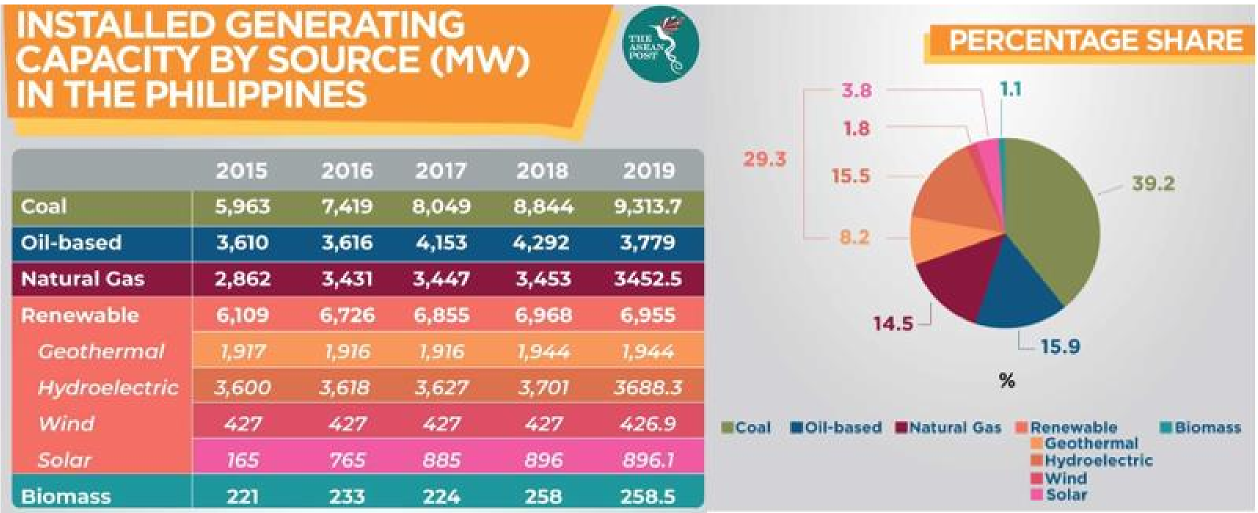
The Sergio Osmeña Bridge was inaugurated on July 4, 1973, it is the first bridge connect Lapu-Lapu city with the Mandaue city, in August 1999, the second bridge, Marcelo Fernan Bridge was completed to decongest the traffic from the older Sergio Osmeña Bridge. As the increase of the trip between 2 sides, the traffic volume on first bridge is about 43 thousand vehicles per day and the second is 48 thousand vehicles per day, the bridges are facing the challenge of the traffic, to improve this situation, Cebu city proponents a new bridge that will link the islands of mainland Cebu and Mactan, is targeting complete it by 2028.

1.2.3 Energy structure of Philippine

Philippine is an emerging economy, and the industrial structure is shifting to the industry-based economy from agriculture-based economy, the demand of energy is increasing. Philippine is in the Pacific Ring of Fire, that means it has a high geothermal potential. [18]

In terms of energy use, according to the 2011 primary energy consumption of Philippine, conventional fossil fuel is the main source for its primary energy demands, the energy structure is shown as follow: oil (31%), coal (20%), geothermal (22%), biomass (12%), hydro (6%), wind/solar/biofuel (1%). [18]

In terms of electricity generation, geothermal energy fulfills 41.4% demands of electricity, 28% by coal, 11.4% by hydro, 15% by natural gas and 0.1% by wind, solar and biofuel. [18]

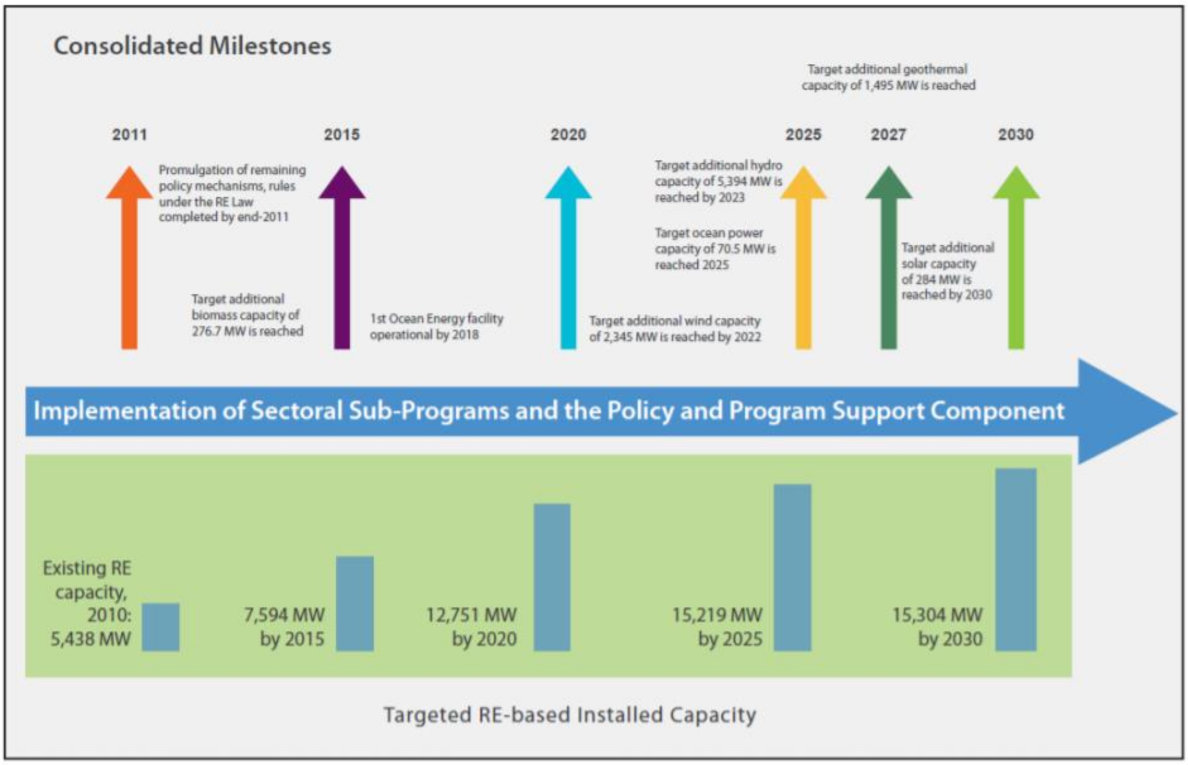


INSTALLED GENERATING CAPACITY BY SOURCE (MW) IN THE PHILIPPINES

	2015	2016	2017	2018	2019
Coal	5,963	7,419	8,049	8,844	9,313.7
Oil-based	3,610	3,616	4,153	4,292	3,779
Natural Gas	2,862	3,431	3,447	3,453	3452.5
Renewable	6,109	6,726	6,855	6,968	6,955
Geothermal	1,917	1,916	1,916	1,944	1,944
Hydroelectric	3,600	3,618	3,627	3,701	3688.3
Wind	427	427	427	427	426.9
Solar	165	765	885	896	896.1
Biomass	221	233	224	258	258.5

Figure 1.15 Installed generating capacity by source in Philippine [19]

In recent years, the number of thermal power plant in Philippine has increased, which causing the percentage of the renewable energy decreased from 32.5% in 2015 to 29.3%, which was not favorable for the target of sustainability, on 27 October 2020, Philippines has announced the suspension of new coal-fired thermal power plants. Suspension of permits for new coal projects could greatly reduce the country's dependence on fossil fuels.



Note: The National Renewable Energy Program (NREP) is a live document and will be subjected to public consultations. Figures presented may change based on regular updates of the NREP.

Figure 1.16 Consolidated RE roadmap

The Philippine's government also set the target for RE-based energy capacity installation with detailed target capacity number every 5 years start from 2011 and aimed to the capacity 15304 MW for 2030 (**Figure 1.16**). [21]

Feed-in tariff subsidies, also known as mandatory feed-in tariff subsidies, renewable energy repurchase tariffs, protective classified tariff systems, or government power purchase systems, are a policy mechanism aimed at accelerating the widespread application of renewable energy. The government signs a long-term contract with individuals or companies that use renewable energy to generate electricity. During this period, for every kilowatt-hour of electricity delivered to the public grid, the generator can earn a number of subsidies in addition to the original electricity price.

RE Technology	FIT Rate ($/kWh)	Degression Rate	Installation Target (MW)
Wind	0.18	0.5% after 2 years from effectivity of FIT	400*
Biomass	0.144	0.5% after 2 years from effectivity of FIT	250
Solar	0.21** (0.188)	6% after 1 year from effectivity of FIT	500*
Run-of-River Hydropower	0.128	0.5% after 2 years from effectivity of FIT	250
Ocean	Deferred		10
			1,410 MW

* Amended solar energy installation target from 50 MW to 500 MW and 200 MW to 400 MW for Wind.

**$0.21 is effective until March 15, 2015, while new rate $0.188 was approved by ERC effective until March 15, 2016

Note:
Assumption; $1=Php 46.00

Figure 1.17 Feed-In Tariff (FIT) Rates in Philippine

Philippine's government has introduced the Feed-In tariff for the different type of renewable energy includes wind, biomass, solar, hydropower and ocean energies, the several policies introduced is aimed to encourage energy producers or individual increase the renewable energy investment and usage. [21]

1.2.4 Section Summary

Cebu's economy has risen to become one of the fastest growing in Philippine thanks to its development of the outsourcing industry and tourism, but the infrastructure hasn't kept up with this growth, the actual transit system of Cebu is not able to fulfill the increasing transport demand of people.

The traffic problem of Metro Cebu is mainly caused by the fast urbanization and population growth in Philippine, but also caused by the shortage of traffic management, road infrastructure and public transportation system, as mentioned in section **1.1 (The impact of traffic on the environment),** the traffic congestion causing the increase of the emission and consumption from the vehicle, the traffic pollution has become the main reason of the degradation of the air quality in Metro Cebu. With the development and spread of electric vehicle, the on-road pollution can be reduced, but it brings new challenge to the electric power system and electric supply system, to get enough benefit from the development of electric vehicle, the adjustment of energy structure and introduce the renewable energy are important.

1.3 Target of Case Study and Structure of Thesis

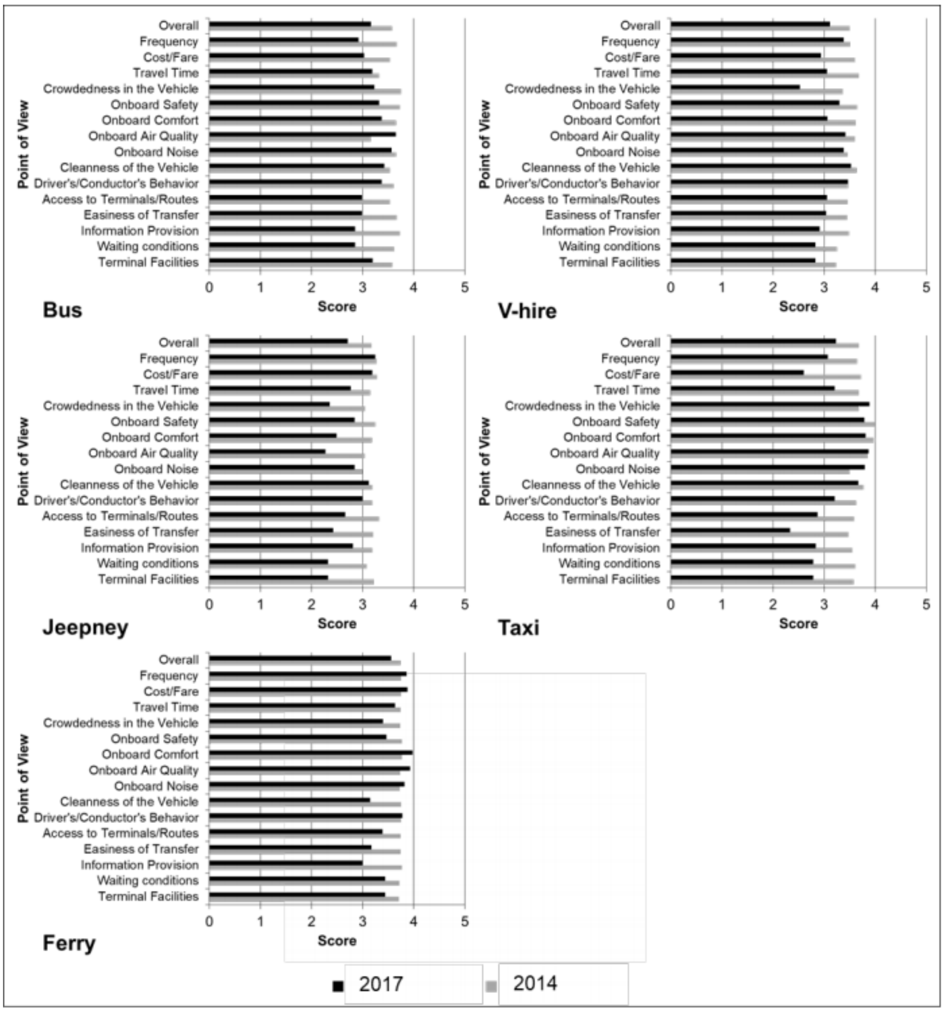
This thesis belongs to a greater case study project, in our case study, the focused area is the region near to the Mactan-Cebu international airport and positioned in Lapu-Lapu city. This project group is a multi-disciplinary group, its member includes students from civil engineering (Daniele Iunti) and from automotive engineering (Ma Fanshu and Zhang Zheyuan).

1.3.1 Problem Definition

Year	Domestic	International	Total	Change
2005	2,106,380	672,284	**2,778,664**	▲ 6.39%
2006	2,291,952	778,210	**3,070,162**	▲ 10.49%
2007	2,765,523	965,977	**3,731,500**	▲ 21.54%
2008	2,997,161	994,089	**3,991,250**	▲ 6.96%
2009	3,841,990	920,913	**4,762,903**	▲ 19.33%
2010	4,206,651	1,206,801	**5,413,452**	▲ 13.66%
2011	4,748,333	1,467,613	**6,215,946**	▲ 14.82%
2012	5,257,941	1,513,377	**6,771,318**	▲ 8.93%
2013	5,369,929	1,626,183	**6,996,112**	▲ 3.32%
2014	5,160,109	1,679,740	**6,839,849**	▼ 2.23%
2015	5,769,104	2,012,135	**7,781,239**	▲ 13.76%
2016	6,334,283	2,436,355	**8,770,638**	▲ 12.72%
2017	6,904,978	3,145,962	**10,050,940**	▲ 14.60%
2018	7,611,398	3,788,489	**11,377,887**	▲ 13.20%
2019	8,370,466	4,291,589	**12,662,055**	▲ 11.29%

Figure 1.18 Passenger movements of Mactan-Cebu airport [22]

As mentioned in previous section. The Mactan-Cebu airport is the second busiest airport in Philippine, the fast development of Metro Cebu has increased the passenger flow of the airport, the passenger number of 2019 is 4.5 times with respect to the number in 2005, this increasing of passenger flow put pressure on the traffic in the surrounding areas, especially to the Lapu-Lapu city where the airport is located. As the only airport in the province Cebu, the Mactan-Cebu airport also be charged with the passengers towards other cities in province Cebu, because of that, the majority passenger's destination is not internal in the Lapu-Lapu city but are the cities at main island of Cebu province, the two bridges connect Mactan island with the mainland Cebu became the must-pass way toward other side.



Source: JICA Project Team

Figure 1.19 Assessment of Current Level of Public Transport Services [4]

From the survey about satisfaction level of public transport services made by the JICA project team, for majority criteria, the grade obtained in 2017 is lower than that of 2014, which exactly proved that the transportation system existing in Metro Cebu cannot fulfill the fast-growing traffic demand of people. In Lapu-Lapu city, there are many intra city transportation demand is met by the tricycle. The figure below is the traffic volume by road link with the unit VCR, VCR is the traffic volume to capacity ratio, can qualitatively represent the road stress by traffic.

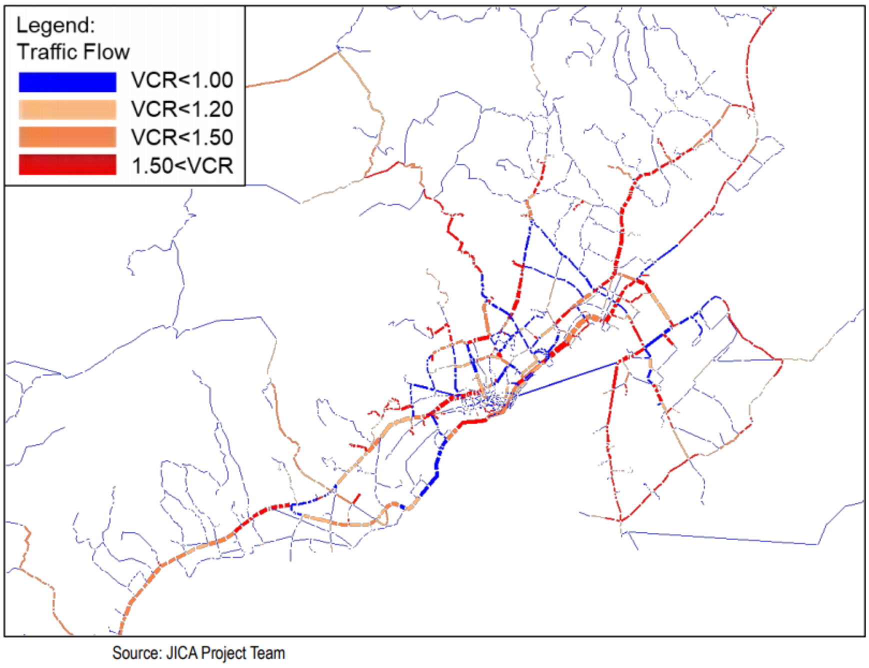


Figure 1.20 Traffic Volume by Road Link[4]

The **Fig. 1.20** shows the stress on the circumferential road of the island is quite high and for the road segment between the entry intersection of two bridge, the road stress is lower.

1.3.2 Structure of thesis and target of case study project

As the transport transfer station of the Metro Cebu, Lapu-Lapu city must increase it inter city transportation, makes the passengers output from the Mactan-Cebu airport reach the mobility center easily, from the intra city point of view, due to the insufficient of the public transportation system, more choice of the transport method needs to be equipped.

Our project target is to reduce the traffic pollution and the traffic congestion of the city by propose a mobility hub in an important point of the city, a moving walkway will connect the mobility hub and the airport in order to give the passengers a choice to reach the center of the city and to reach the entry of the bridge easily, the mobility hub includes a smart parking system equipped with the charging pile for the electric vehicle, an share micro-mobility center equipped with the battery swap station and a moving walkway with multi conveyer, the solar panel will be installed on the roof of the mobility hub but also on the corridor of the walkway in order to provide the renewable energy based electricity to the electric mobility devices required by the mobility hub, a digital twin system will be implemented for the better planning, managing and maintaining of the area, more detailed description of the project will be shown in **Chapter 3 Methodology.**

In the **Chapter 2 State of the Art**, the state-of-the-art of smart city and smart transportation solutions will be discussed, the concept and application of the urban and traffic digital twin will be presented, and the VR application in the smart city will be mentioned.

Chapter 3 Methodology will include the overall view of the whole project, the specific description of the workflow of a methodology to set up a microscopic traffic simulation based on VISSIM using the data collected from the real world, the workflow of preparing the digital model of the urban area efficiently using 3DS MAX and CityEngine, the method to build the interactive driving environment based on the Unity and its data exchange with the VISSIM, and finally, the VR application to the driving simulator in order to obtain a more immersive experience in order to help the project planning and infrastructure testing will be described.

Chapter 4 Conclusions and Future Development describes the results and possible improvement of the project and also for the driving simulator and traffic scene research.

2. State of the Art

Informatization theory was first proposed by Daniel Bell (1959) that informatization refers to the process of benefiting the society from cultivation and development of the new productivity represented by computer-based intelligent tools. [23] “informatization” and “information society” have been received extensive research and attention in the middle and late epochs of seventies, people have realized that informatization is related to the survival and development of society and organization, it plays an irreplaceable role in improving efficiency, increasing benefits, and obtaining competitive advantages.

As information technology continues to make major breakthroughs in the 21st century, informatization has further been widely used and penetrated in various fields of society. Urban informatization is a process of urban development characterized by information at a certain historical stage. The urban social structure has evolved from a development process centered on material and energy to information and knowledge, in this long-term development process, new information and communication technologies have been continuously adopted and applied to various departments and industries in the city, which has improved the management efficiency of the city. [24]

2.1 Smart City and Smart Transportation

With the acceleration of the world’s urbanization process, while enjoying a high quality life, people begin to pay more attention to the quality and comfort of life. The large population, crowded housing, tight traffic, environmental pollution, ecological deterioration, etc. have seriously affected the quality of people’s lives, increased the difficulty of urban management, and caused a series of social, economic and management problems. Driven by the global informatization trend, in 2009, IBM took the lead in proposing the new concept of “smart city”. Smart cities have quickly become a strategic choice for urban informatization worldwide and are used by countries all over the world to respond to the international financial crisis and seize future technology, smart city has become a new model of future urban development. [24]

2.1.1 Smart City

The core idea of a smart city is to use new generation of information and communication technology with cloud computing and the internet of things as the core, based on people-oriented concept to change the way people, organization, government, and city interact with each other, in order to respond to various needs quickly and intelligently, thereby improving the efficiency of urban management and providing better services to residents. [24][26]

From the perspective of the functions of smart cities, the framework structure of smart cities can be divided into four levels: perception layer, network layer, data layer and application layer (**Fig. 2.1**) [27]

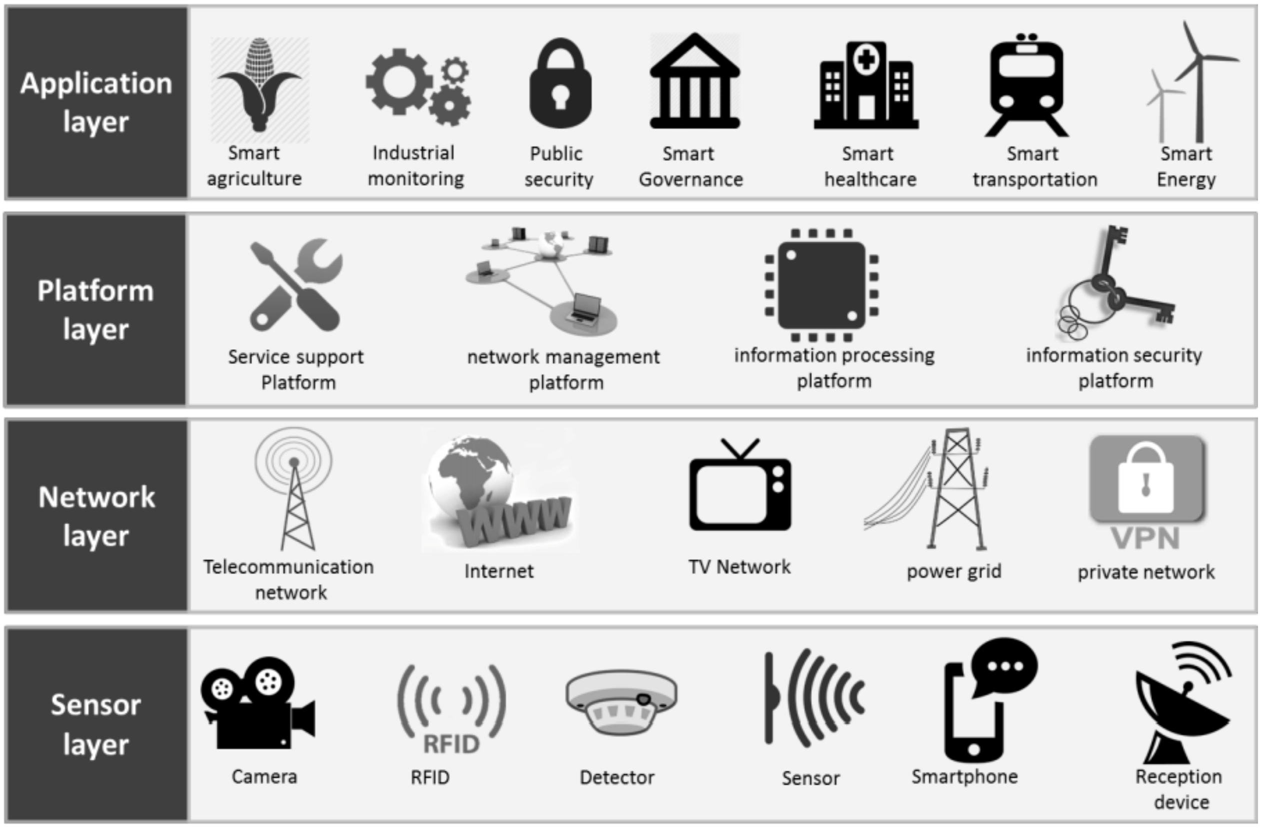


Figure 2.1 Infrastructure Layer of Smart City [28]

The functions of each level are thorough perception, extensive interconnection, massive data storage and deep intelligence. Among them, thorough perception refers to the use of equipment (different sensors), systems or processes that perceive, capture and transmit information anytime and anywhere to quickly obtain various information in the city, extensive interconnection refers to the use of various forms of high-speed communication network to connect the information system of individual, government and enterprise, let them interact and share the information perceived from each end in order to complete the activities in collaboration way. Massive data storage refers to the data warehouse and data mining technologies to classify and manage a large volume of scattered data, from a global perspective analyze and solve the problems in real time. Deep intelligence refers to deep analysis of the information resources obtained and solve the specific issues in a more innovative and comprehensive way to better support urban development decisions and actions. Smart city has wide application areas, this area cover fields such as transportation, public utilities, education, health and social care, public services etc. [29]

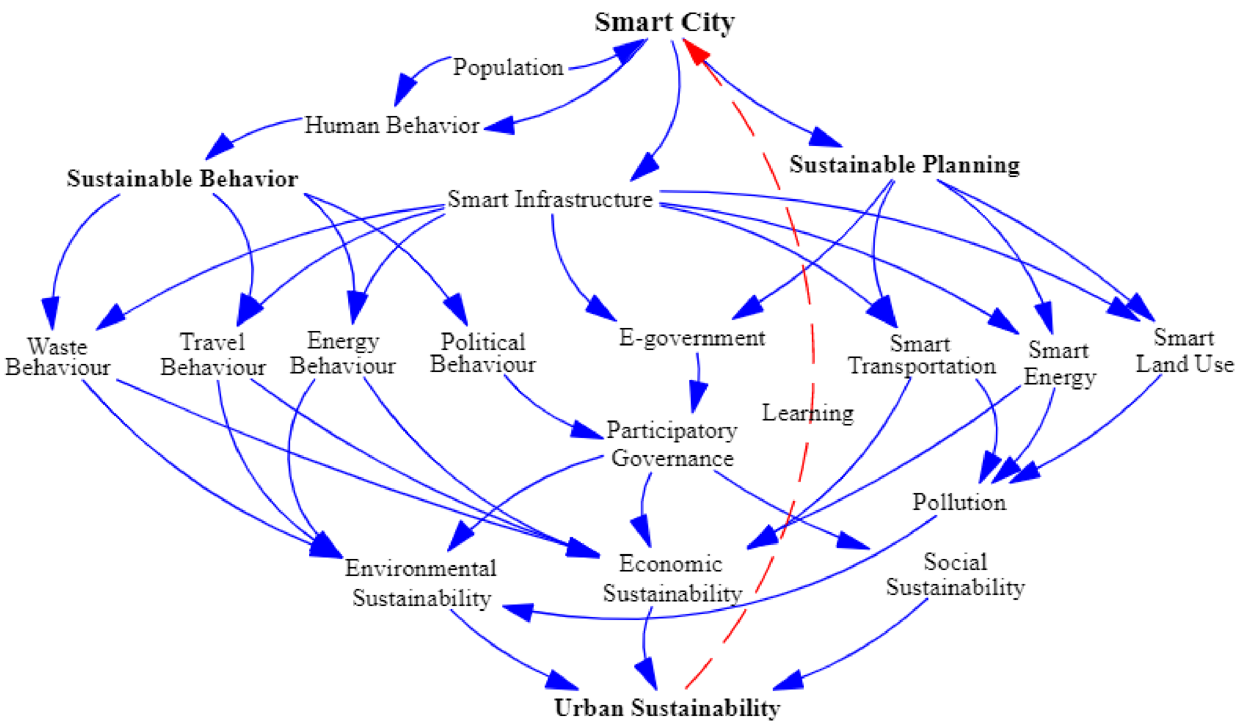


Figure 2.2 Dynamic impact on the urban sustainability from smart city [30]

To achieve urban sustainability, both citizen behavior and official decision making need to be more efficient and sustainable [30]. The essence of a smart city is to provide accurate information at the right time so that citizens, service providers, and city managers can make better decisions to improve the quality of life of urban residents and the overall sustainability of the city. Smart cities encourage sustainable behaviors of citizens, by changing their political, energy, travel, and waste behaviors, and promote the structure of sustainable urban planning and urban management by changing urban infrastructure (energy, land use, and transportation systems). [31][32]

2.1.2 Smart Transportation and its applications

The problem of traffic congestion is an important link in urban development, it’s a key content of government public governance, and a topic vigorously discussed by scholars and research institutions. In the development of smart cities in full swing, the smart development of transportation is also one of the important goals to be achieved. The smart transportation system aims to use various technological innovations in the transportation field and use the penetration and diffusion of high-tech to promote the transformation of transportation management methods, realize the modernization of transportation governance capabilities, and ultimately realize the green, healthy and sustainable city development. [31][32]

Smart transportation is developed under the support of a new generation of information and communication technology. The effective operation of smart transportation is inseparable from the comprehensive utilization and analysis of the internet of things, big data, cloud computation and artificial intelligence. Deploy the intelligent hardware of urban road traffic through the internet of things, lay the intelligent big data traffic system, and use internet technologies to realize the whole process of visual monitoring and digital monitoring. Establish an urban traffic brain, through the use of human-like brain's comprehensive intelligence such as perception, coordination, learning, control, decision-making, feedback, and innovation to acquire and analyze the urban traffic related information of the city, make the comprehensive research and judgement, generate the countermeasures for the specific problem. This whole process will help all traffic participants to fully and accurately understand traffic information and make precise decisions, so as to realize the overall optimization of urban traffic management and services, solve the problem of increasing urban commuting costs, and effectively alleviate urban traffic congestion. [32]

There are many technics has been developed and adopted in smart transportation field such as real-time traffic information-based navigation system, computer vision-based vehicle information extraction, Green Light Optimized Speed Advisory (GLOSA) system, real-time data based public transportation schedule system, real-time traffic data-based traffic light timing adjustment system etc.

Figure 2.3 Dynamic Path Routing

Based on the traffic data extracted from the sensors (Camera, mobile, inductive loops etc.) find the optimal path not only based on the minimum distance, but the global optimal take in consideration the traffic condition of each possible routes, in order to balance the traffic stress between the roads and avoid the congestion. [33][34][35]

Figure 2.3 Vehicle information extraction [36]

Based on the camera sensor and computer vision technology, extract the information of the vehicle in the real-time video stream or picture, and identify the relevant information of the specific details of the vehicle, license plate, etc. and send this data to the traffic management center in order to better manage the traffic and the illegal driving behavior.

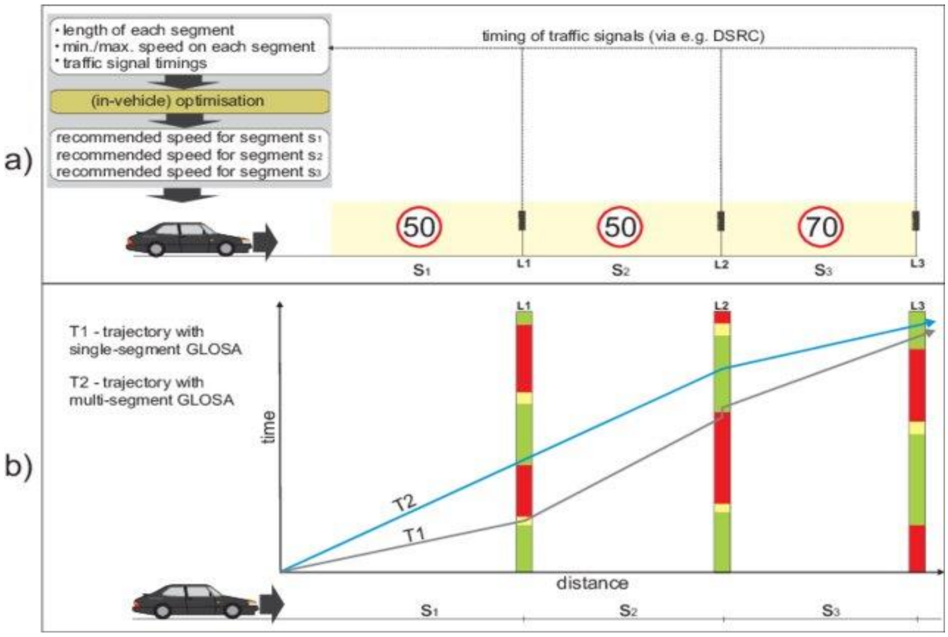


Figure 2.4 Green Light Optimized Speed Advisory (GLOSA) system [38]

Based on vehicle to x (VtoX) communication technology [37], the traffic signal can communicate with the vehicle around wirelessly, and advise the driver in which speed can meet the green light without stop and waiting depending on the light switch time and distance between vehicle and traffic signal, with this technology, the traffic delay can be reduced and the emission decreased thanks to the lower idling time of vehicle. Additionally, the traffic signal can be equipped with a camera and computer vision chip, through the edge computation technic, the traffic signal can smartly adjust its signal timing depends on the traffic situation, vehicle priority, weather etc. without intervention of the human.

The development of smart transportation system is a revolution in the history of human transportation, it expands the viewport of the traffic participants from surrounding area to the urban level, helps people obtain and process massive traffic data, and continuously learns and iterates from historical data to make the future traffic predictions more accurate.

2.2 Digital Twin

The idea of digital twin appeared for the first time in professor Grieves's product life cycle management course at the university of Michigan. However, at that time, the term "Digital Twin" had not been formally proposed. Grieves called this idea "Conceptual Ideal for PLM (Product Lifecycle Management)" (**Fig. 2.5**)

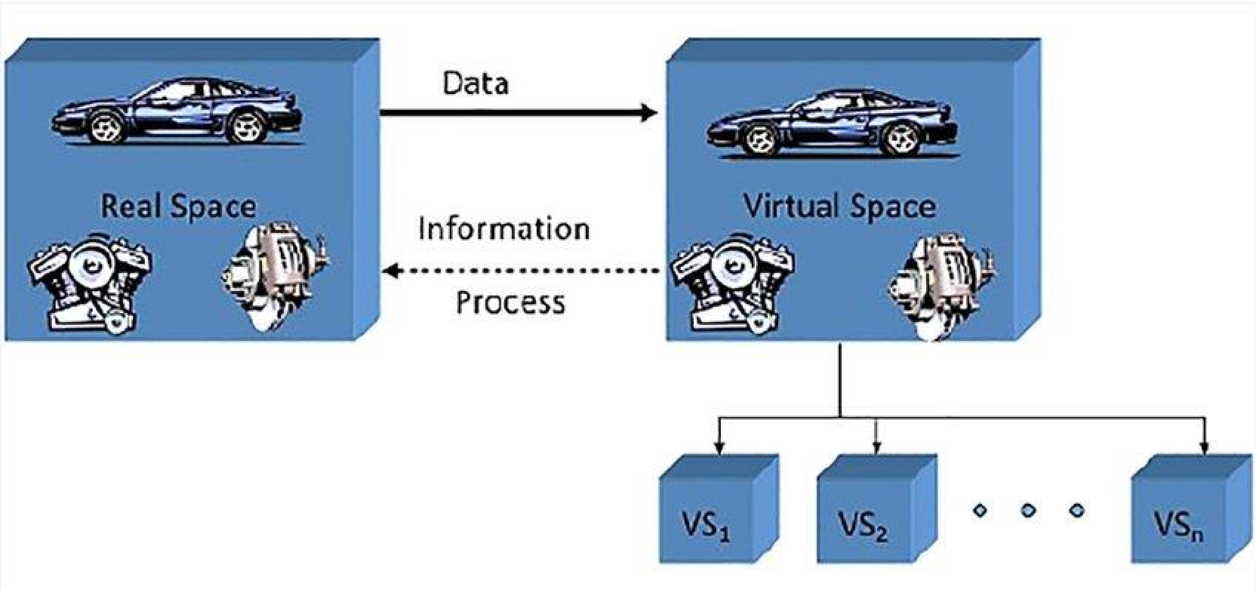


Figure 2.5 Conceptual ideal for PLM [39]

The basic idea of the digital twin has been reflected in this vision, that is, the digital model constructed in the virtual space and the physical entity are interactively mapped to faithfully describe the trajectory of the physical entity's entire lifecycle. [39][41]

Until 2010, the term "Digital Twin" was formally proposed in NASA (National Aeronautics and Space Administration)'s technical report and defined as "a system or aircraft simulation process that integrates multiple physical quantities, multiple scales, and multiple probabilities". In 2011, the US Air Force explored the application of digital twins in aircraft functional management and discussed in detail the technical challenges of implementing digital twins. In 2012, NASA and the US Air Force jointly published a paper on digital twins, pointing out that digital twins are one of the key technologies to driving the development of future aircraft. [42] In the next few years, more and more research applied digital twins to the aerospace field, including airframe design and maintenance, aircraft capability evaluation, aircraft failure prediction, etc. [40]

In recent years, digital twins have become more and more widespread. At the same time, thanks to the development of new-generation information technologies such as the Internet of Things, big data, cloud computing, and artificial intelligence, the implementation of digital twins has gradually become possible. At this stage, in addition to the aerospace field, digital twins are also used in industries such as power, shipbuilding, urban management, agriculture, construction, manufacturing, oil and gas, health care, and environmental protection. Especially in the field of intelligent manufacturing, digital twins are considered to be an effective means to realize the interactive integration of the manufacturing information world and the physical world. Many well-known companies (such as Airbus, Lockheed Martin, Siemens, etc.) and organizations (such as Gartner, Deloitte, China Association for Science and Technology, etc.) have attached great importance to digital twins and have begun to explore new smart production models based on digital twins. [40]

2.2.1 Definitions and Characteristics of Digital Twins

Different field, academy, organization, company has different ways to define digital twins, (Tao et al. 2018) has defined: digital twin is a virtual entity that creates a physical entity in a digital way, with the help of historical data, real-time data, and algorithm models, it is a technical means to simulate, verify, predict, and control the entire life cycle of a physical entity. [44] (Aaron et al.2017) defined digital twin as the evolving digital data of the historical and current behavior of physical objects or processes that help optimize business performance. The digital twin model is based on large-scale, cumulative, real-time, real-world data measurement across a series of dimensions. [45]

GE (General Electric company) think digital twins are software representations of physical assets and processes that are used to understand, predict, and optimize performance to achieve improved business results, Digital Twins can help customers across three core areas: Asset, Network and Process. [43] The digital twin company has established itself in the industry, and it has revolutionized processes throughout the value chain. As a virtual representation of a product, production process or performance, it enables seamless linking of various process stages, this can continuously improve efficiency, minimize failure rates, shorten development time, and open new business opportunities, in other words, it can create a lasting competitive advantage. [46]

From previous definitions of different field and company, the following Characteristics of digital twin can be observed.

Interoperability: The physical objects and digital space in the digital twin can be mapped, dynamically interacted, and connected in real time. Therefore, the digital twin has the ability to map physical entities with various digital models, and can convert, merge, and establish "expressions" between different digital models.

Scalability: Digital twin technology has the ability to integrate, add and replace digital models, and can be expanded for multi-scale, multi-physical, and multi-level model content.

Real-time: Digital twin technology requires digitization, that is, managing data in a way that can be recognized and processed by a computer to characterize physical entities that change along the time axis. The represented objects include appearance, state, attribute, and internal mechanism, forming a digital virtual body mapping of the real-time state of physical entities.

Fidelity: The fidelity of the digital twin refers to the closeness between the digital virtual model and the physical entity. It is required that virtual bodies and entities not only maintain a high degree of simulation of the geometric structure, but also simulate in terms of state, phase, and time. It is worth mentioning that in different digital twin scenarios, the degree of simulation of the same digital virtual body may be different. For example, in a working scenario, it may only be required to describe the physical properties of the virtual body without paying attention to the details of the chemical structure.

Close loop: The digital virtual body in the digital twin is used to describe the visual model and internal mechanism of the physical entity, so as to monitor the state data of the physical entity, analyze and optimize the process parameters and operating parameters, realize the decision-making function, and give the digital virtual body and physical entity a brain, so the digital twin has a closed loop. [44][40]

2.2.2 Digital Twin Framework and its Application

As mentioned before, the digital twin consists of physical entity in real space, digital virtual entity in virtual space, and the connection of data and information that ties them together (**Fig. 2.6**). [47]

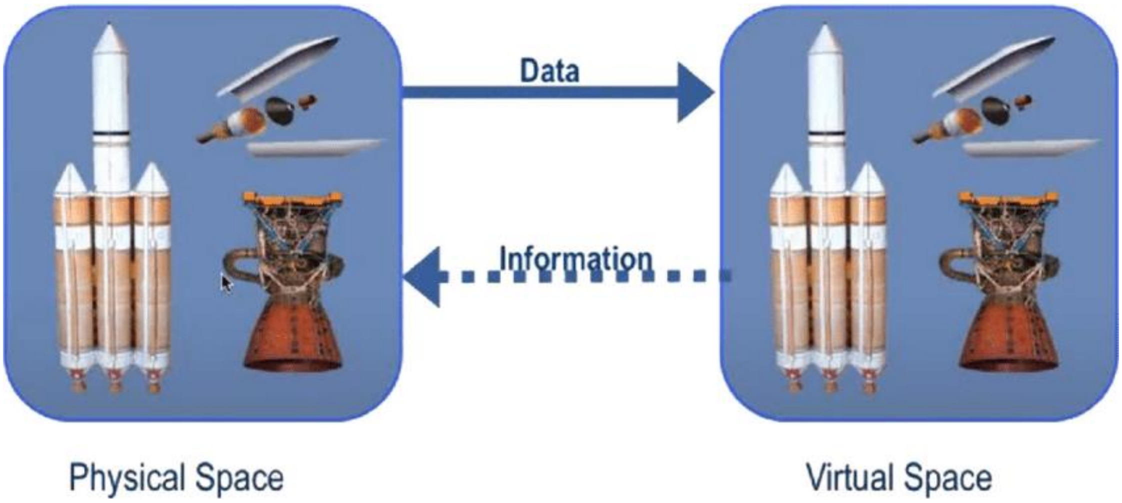


Figure 2.5 The digital twin model proposed in [47]

Grieves points out the most powerful tools in the human's knowledge toolkit supported by digital twin are conceptualization, comparison, and collaboration.

Where **conceptualization** means that instead of transforming the information from the physical word to the symbols, numbers, letters, and then re-conceptualize it visually, digital twin can let us directly see the situation and eliminate the inefficient due to the information loss during the information transformation.

Comparison means that if the virtual product and real product is separate, then the comparison is inefficient because we must find the corresponding information separately and then work out difference, with the digital twin, we can view the ideal characteristic, the tolerance corridor around that ideal measurement to avoid the result unacceptable.

Collaboration means the digital twin can break the limitation of the location, the situation of physical model can be perceived and controlled by the participants wherever they are. [47]

"Twin" does not mean the Digital Twin does not exist until and unless there is a physical entity, because the digital twin framework should cover the entire product lifecycle and is especially useful during the planning and creation phase. [48] The operation process can be estimated through digital twin before it be produced in order to reduce the waste of energy, material, and time of development and operation.

To further apply the digital twin in more fields, the Beihang digital twin technology research team has expanded the existing three-dimensional model and added two new dimensions of twin data and services, and creatively proposed the concept of the digital twin five-dimensional model (**Fig. 2.6**). [49]

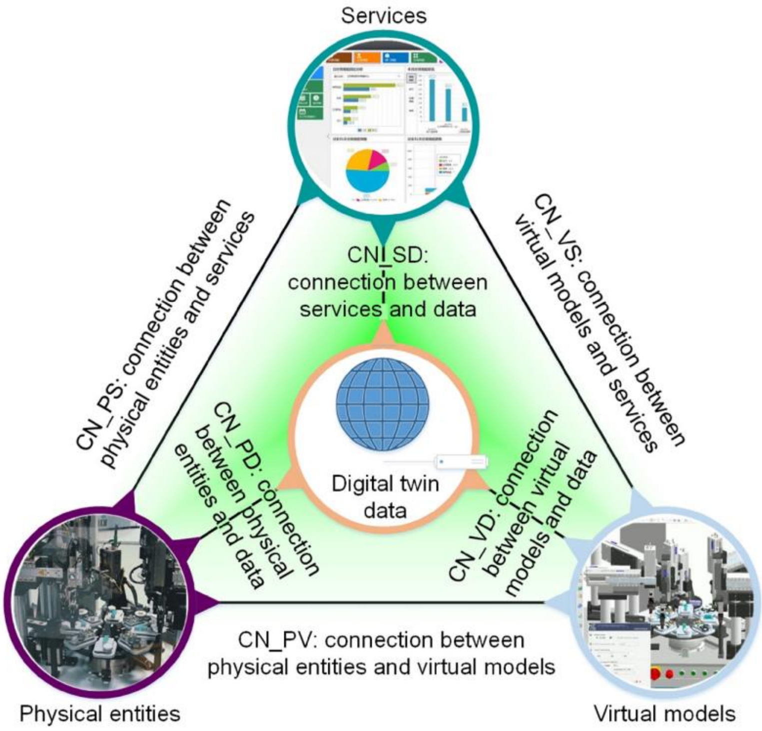


Figure 2.6 Five-dimension digital twin model proposed in [49]

The five-dimension digital twin model includes: physical entities, virtual models, services, digital twin data, and connections. [49][50]

Physical entities are the basis of the five-dimensional digital twin model, accurate analysis and effective maintenance are the prerequisites for establishing the model, it is hierarchical and generally includes unit level, system level and complex system level. The unit is the minimal function realization unit, the system is formed by several unit, these units work together to accomplish certain mission, the complex system is the system of the system, is able to realize the organization and management of various subsystem. [49][50]

Virtual models include geometric models, physical models, behavioral models, and rule models. These models can describe and characterize physical entities from multiple time scales and multiple spatial scales. Geometric model is a three-dimensional model describing the geometric parameters (such as shape, size, position, etc.) and relationships (such as assembly relationship) of physical entities. It has good temporal and spatial consistency with physical entities. The geometric model can be created using 3D modeling software such as Solidworks, 3D Max, Revit, ProE etc. or using equipment (such as 3D scanner). Physical models add information on the physical attributes, constraints, and characteristics based on geometric model. Behavior models describes the real-time response and behavior of physical entities with different spatial scale under different time scales, creating a physical model's behavior model is a complex process, various models such as problem models, evaluation models, decision models etc. are involved. Rule models includes laws and rules based on historical linked data, experiences based on tacit knowledge, and standards and guidelines in related fields. These rules self-growing, self-learning, and self-evolving over time, so that virtual entities have real-time judgment, evaluation, optimization, and prediction capabilities. [50]

Services refers to the service-oriented packaging of various data, models, algorithms, simulations, and results required in the application process of the digital twin, includes the "functional services" that supports by the operation and realization of the internal functions of the digital twin in the form of tool components, middleware, and module engines and "Business service" that meets the different business needs of different users in different fields in the form of application software, mobile app, etc. Among them, functional services provide support for the implementation and operation of business services. [50]

Digital twin data is the driver of the digital twin, it mainly includes physical entities data, virtual entities data, services data, knowledges data, and fusion data. Physical entities data mainly includes physical element attribute data which reflecting its specifications, functions, performance, relationships, etc. and dynamic process data reflecting its operating conditions, real-time performance, environmental parameters, sudden disturbances, etc., which can be collected through sensors, embedded systems, and data acquisition cards. Virtual entities data mainly includes virtual entities related data, includes the data related to geometric models, physical models, behavioral models, and rule models. Services data includes the data related to functional services and business services. Knowledge data includes expert knowledge, industry standards, rule constraints, reasoning inferences, common algorithm libraries and model libraries, etc... Fusion data is obtained by transforming, processing, analyzing, and combining the data previously mentioned. [50]

Connection is the communication and interconnection between the modules previously mentioned. [50]

Based on the above research on the five-dimensional digital twin model, the article [50] further explores the use of the five-dimensional digital twin model in satellite/space communication networks, ships, vehicles, power plants, aircraft, complex electromechanical equipment, three-dimensional warehouses, medical care, manufacturing workshops, and smart cities. Application in 10 areas (**Fig. 2.7**)

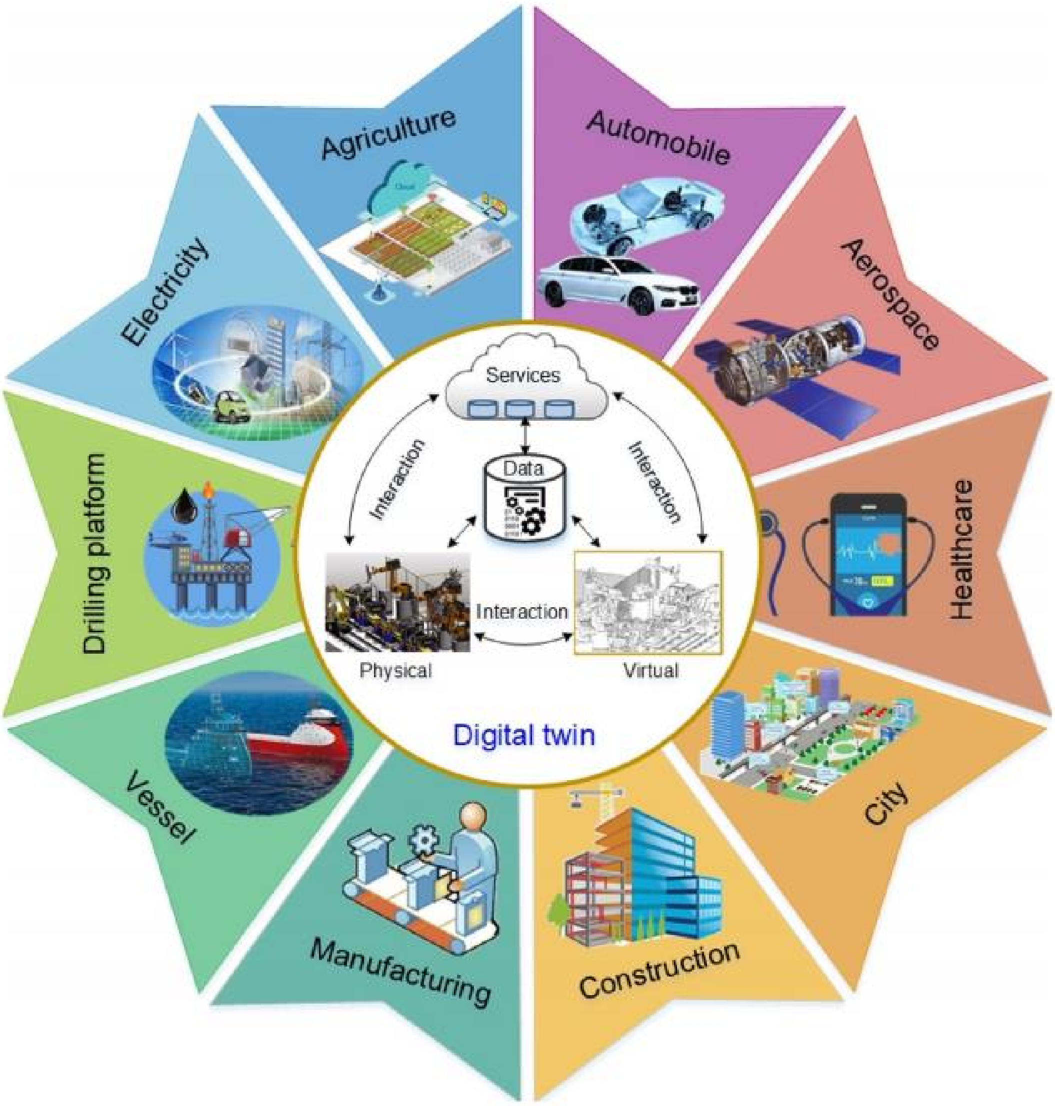


Figure 2.7 Ten application areas of Digital Twin discussed in [50]

2.2.3 Digital Twin and Smart City

A city is an open, huge, and complex system with the characteristics of high population density, dense infrastructure, and coupling of subsystems. How to realize real-time monitoring of various data information of the city, and efficiently manage the city around the top-level design, planning, construction, operation, safety, people's livelihood, and other aspects of the city, which is the core of modern city construction. [50]

From the 5-dimension digital twin model point of view, Tao etc. [50] points out the urban digital twin could include physical city, virtual city, city's big data, the interconnection between virtual and physical, and intelligent services. Physical city is the urban area in the real space, its our target of improvement and service from the implementation of the digital twin, Through the deployment of sensors in the sky, ground, underground, river and other levels of the city, it can fully perceive and dynamically monitor the city's operating status. The virtual city is the digital models built based on the physical city, the virtual city can simulate the behavior of the people, objects, transportation, environment etc. of the real city under real environment. City's big data is to converge the data from various traces of urban infrastructure, transportation, environmental activities, virtual city simulations, and various smart city service records, etc. into urban big data to drive the development and optimization of digital twin cities. The interconnection between physical and virtual world means that the activities such as urban planning, construction are not only existing in the physical space, but also greatly expanded in the virtual space, the interaction, collaboration, and integration of virtual and realty will define a new model of urban development in the future. Intelligent service means through digital twins, the city is planned and designed to guide and optimize the municipal planning, ecological environment management, and traffic control of the physical city, improve citizen services. [49][50]

Sergey Ivanov etc. [51] points out some information source from the physical city and the relevant possibilities of digital twin's system based on them.

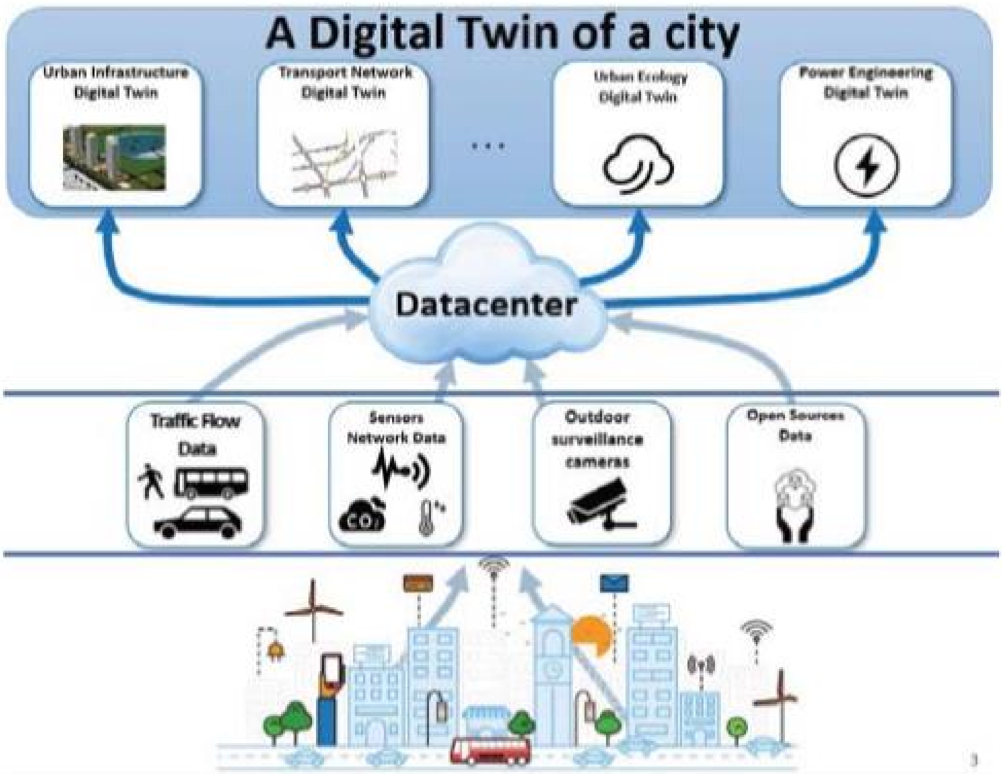


Figure 2.8 Example of the Interface's appearance in [51]

Where the data source includes the information from the traffic of private, commercial and public transport, also the congestion area collected through different sensors and methodologies, the information of the physical parameter from various sensors installed in the city, such as temperature, humidity, noise, radiation, particular matter pollution in the air, density of the CO2 etc., the information from the surveillance cameras, the camera video captured can contain the information which is hard obtained from other sensor or methodology such as to understand if a freeway under the traffic congestion, but this information should be extract from video using some technology such as image processing, the information from the open sources such as the meteorological conditions, open reporting info of companies etc.[51]

From the data above mentioned, the digital twin of urban infrastructure, the digital twin of transport network, the digital twin of urban ecology, the digital twin of power engineering etc. can be supported [51]

LI Deren etc. [52] proposed a layer-based structure of the urban digital twin (**Fig. 2.9**)

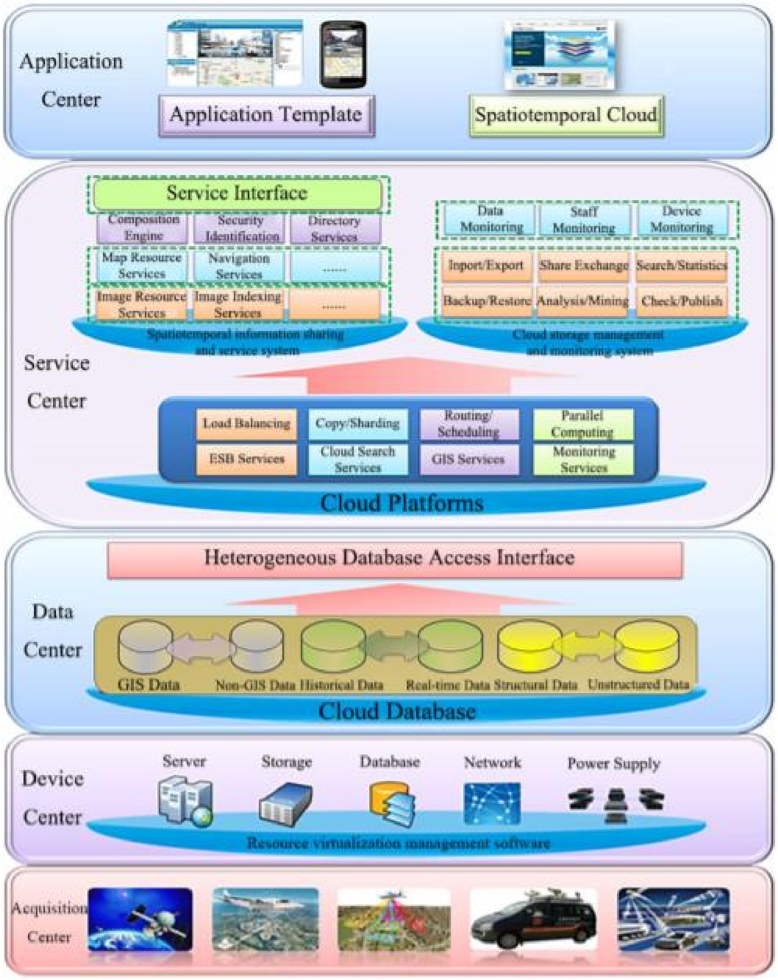


Figure 2.9 The structure of a smart city public information cloud service platform based on digital twins [52]

The platform is composed of infrastructure layer, software development and operation platform layer, and application layer. [52]
In summation, Digital twin technology is an effective technical means to realize smart cities. With the help of digital twin cities, it can improve the quality and level of urban planning, promote urban design and construction, assist urban management and operation, and make urban life and environment better.

2.2.4 Digital Twin and Smart Transportation

Simulating the urban traffic, running out of the optimal solution, and then improving the actual traffic situation based on the simulation results is one of the important functions of urban traffic digital twin technology. Applications such as mid-to-micro simulation and industrial simulation driven by digital twin technology are increasingly permeating into the research and development of traffic management, smart transportation, and autonomous driving technologies, and will play an important role in future urban transportation.
Li etc. [53] points out that let things that require high costs or difficult to achieve in the real world be quickly implemented in the virtual world. This is the meaning of digital twins, for example, it is very difficult to find solutions for the traffic congestion problem in big cities, modify roads in the real world or do the field test are very expensive and time consuming. In the scenario shaped by digital twin technology, hundreds of tests can be made, let every car, every road, and even many roads line designs and steering designs be tested in the simulator, run the optimal solution, and then return to the real world to implement, and huge amount of sensor data can be synchronized to the digital twin system in real time, after the data is correlated, vehicle-to-infrastructure communication and vehicle-to-vehicle communication can become simpler.
The report of “We transport” [54] smart transportation project mentioned that smart devices such as car-side sensors, roadside cameras and even mobile phones collect a large amount of traffic data every day, such as the number of lanes, slopes, traffic flow, traffic light positions, etc. the collected data is transmitted to the cloud data management center through the communication network, perform data classification, management, cleaning, labeling, etc. to form effective data resources, then the intelligent engine analyzes, mines, retrieves, and visualizes the data, and the data can be processed in a flexible and convenient interface such as API/SDK. Based on these large amounts of real data in the real world, model the virtual world as accurately as possible. In the end, the digital twin platform built with massive real data as the "raw material" will build a model of the

physical world in the virtual space, and realize the one-to-one mapping between the digital twin environment and the real environment, but the difference between virtual and real is that the world of digital twins can break through the limitations of time and space, drive and deduct future urban traffic operations with real data, and explore the best solutions for smart city planning and traffic operation management.

From application point of view, the transportation digital twin has following value:

Improve the efficiency of driverless training: The digital twin technology provides a richer and more comprehensive source of test scenarios for autonomous driving test evaluation methods and covers a greater range of multiple test scenarios such as the functional safety of autonomous driving. Through the gradual deployment of smart transportation and vehicle-road collaboration, scenarios derived from the entire life cycle of vehicles can be extracted and applied for testing, such as vehicle operating database, vehicle network operating database, and intelligent transportation database. These scenarios based on historical operating data and are also refined by secondary processing to provide a big data foundation for typical test scenarios such as edge working condition case libraries and human-computer interaction evaluation case libraries. [40][55]

Traffic accident analysis: Digital twin technology has great application value in traffic accident analysis scenarios. It can help trace and analyze the specific causes of accidents and find the responsibility. Once the accident scenario environment and the trajectory of traffic participants can be tracked, we can restore the scene and observe the process of the accident from different angles.

Traffic control: Using digital twin technology to simulate urban traffic conditions, and then optimize traffic control strategies through evaluation and deduction, this is an important application scenario for digital twins to empower smart transportation. It mainly involves four levels of functions:

One is monitoring and discovery. Through the digital twin system, a closed loop of information acquisition and control can be created, and the whole process can be controlled.

The second is deduction and prediction. After handling the data, it is possible to create micro-behavior models for some participants, and then through the simulation calculation of many traffic participant agents, to obtain macro-simulation results, infer the development of the situation, and realize the prediction function.

The third is to evaluate and optimize countermeasures. Through massively parallel computing, simulation results of many parallel worlds can be evaluated at the same time. Then, through intensive learning and other technologies, we will continue to improve the traffic control plan.

The fourth is historical tracing and review research. When an incident occurs, we can use the digital twin system to recover the entire process of the traffic accident and explore whether each step of the response at that time was done well enough and whether there is space for improvement.

3.3.5 Digital Twin and Virtual Reality

Virtual reality technology (VR) is a computer simulation technology that enables people to enter and experience artificially created virtual worlds in an immersive way. VR has been developed fast in recent years, with the emergence of technologies such as 5G high-speed transmission, the Internet of Things, artificial intelligence, flexible display, and high-performance graphics computing cards, VR technology has spread into commercial field. Now days, applications and equipment based on virtual reality technology have begun to appear in many fields such as education, media, entertainment, medical care, heritage protection, etc.

Figure 2.10 Sensorama

Figure 2.11 Headsight

The earliest VR technology can be traced back to “Sensorama” in 1956 (**Fig.2.10**). It integrates a 3D display, scent generator, stereo speakers, and vibrating seat, and 6 short films for people to enjoy. However, its huge size obstructs it from becoming a commercial entertainment facility.

In 1961, Philco developed a head-mounted display “Headsight” (**Fig. 2.11**). It integrates head tracking and surveillance functions but is mainly used to view secret information.

Figure 2.12 GAFViewMaster

“GAFViewMaster”, which came out in 1966, is the prototype of today's simple VR glasses (**Fig. 2.12**). It achieves 3D visual effects through built-in lenses, but it does not carry any electronic virtual imaging devices or audio equipment.

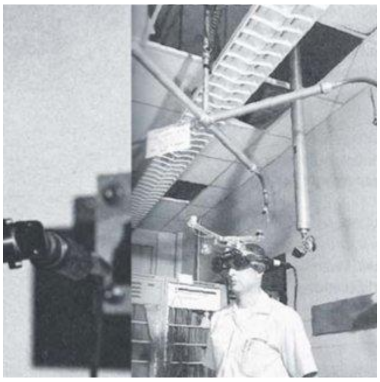


Figure 2.13 The Sword of Damocles

“The Sword of Damocles”, which came out in 1968, is usually considered the true beginning of virtual reality devices (**Fig. 2.13**). It was developed by the Massachusetts Institute of Technology and provided a prototype and reference for the later development of VR and even AR devices. [56] In 1984, the first commercial VR device RB2 was born, equipped with position sensors such as somatosensory tracking gloves. The design concept is almost the same as that of modern mainstream products. In 1985, NASA developed an LCD optical head-mounted display that can provide an immersive experience under the premise of miniaturization and light weight. Its design and structure have since been widely promoted and adopted. In the field of games and entertainment, some well-known companies have also tried to use virtual reality technology to develop related products. In 1993, game manufacturer Sega planned to develop a head-mounted virtual reality device for game consoles but died due to a lackluster response in the internal test. In 1995, Nintendo released Virtual Boy, a game console based on VR technology, but it failed in less than a year because it could only display red and black colors and the game content resolution and refresh rate were low. In 1995, the University of Illinois developed a VR system called "CAVE" to achieve an immersive experience through a three-wall projection space and stereo LCD shutter glasses. The product that truly brought commercial virtual reality technology to the renaissance was Oculus Rift, which came out in 2009. In 2013, it launched an early device for developers at a price of only US$300, representing that commercial VR devices have truly entered the consumer electronics market. [57]

One important goal of the digital twin is to replicate the difficult-to-access asset conditions and the difficult-to-build physical simulation environment to test the best method of project operation or implementation. VR is the easy way to achieve this goal of a digital twin, because the VR technology can provide an immersive environment, that makes people simulate the operation phase in real condition, at the same time, the various input system different from the keyboard and mouse can be implemented with VR such as the Oculus touch to make people feels like operate with the hand in virtual world, also there are other input devices like steering maneuver for VR to simulate the driving process and walk platform to let people has free walk experience in the virtual world.

3. Methodology

As mentioned in **1.3.2 Structure of thesis and target of case study project**, this thesis is belonging to a greater project group which formed by 3 students from automotive engineering and civil engineering, in our case study, the target region is the area in which the Mactan-Cebu international airport located (Lapu-Lapu city), as mentioned in **1.3.1 Problem Definition**, the passenger flow of Mactan-Cebu airport rapidly increased in recent years, and the destinations of many passengers are the main island of province Cebu, which brings traffic pressure to the road network of Lapu-Lapu city. On the other side, focus on the traffic status internal of the city, due to the bad road condition (narrow and non-paved) of the internal area of the Lapu-Lapu city, the public transportation system is inefficient in these areas and cannot fulfil the increasingly traffic demand of citizens. In this chapter, the workflow of case study, and the methodology to realize a digital urban area and its traffic will be described.

3.1 Workflow

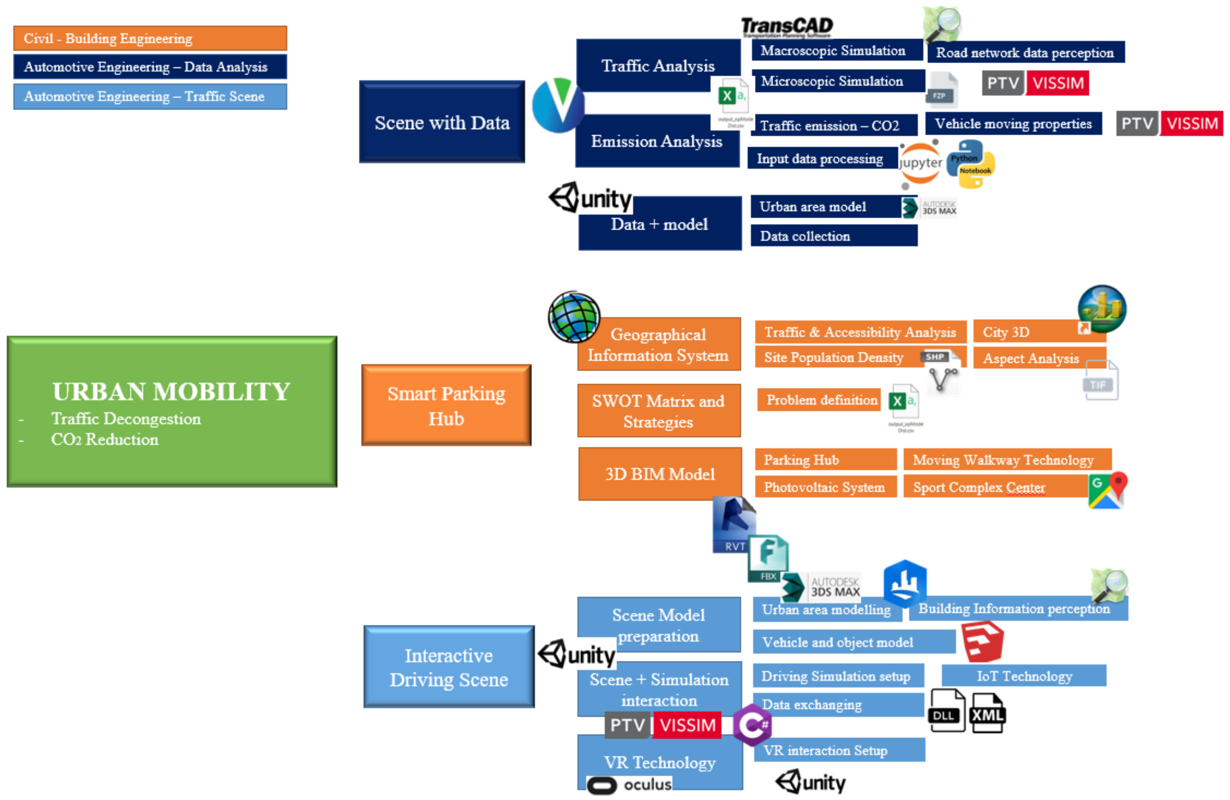


Figure 3.1 Team Workflow

The case study is a multidisciplinary project between civil-building engineering study course and automotive engineering study course. For the civil building engineering field (orange part in **Fig.**

3.1), Daniele is focusing on the design of smart parking hub and moving walkway, problem definition and Geographical Information System (GIS) analysis. For the automotive engineering part, we subdivided it in two directions. Fanshu (dark blue part in **Fig. 3.1**) is focusing on the road network data perception, macroscopic traffic analysis, emission analysis and data visualization. While Zheyuan (light blue part in **Fig. 3.1**) are focusing on traffic scene building, low polygon building batch generation, real time data exchange between software, and VR application to the driving scene.

At the beginning, to have a deep understanding of the site, the civil engineering student Daniele focused on the spatial data management, concerning traffic and accessibility analysis, site population density and the 3D as build of the city by using GIS technology. Simultaneously, the automotive engineering student Fanshu focused on the macroscopic data analysis about traffic by using traffic analysis tool TransCAD and data source from OpenStreetMap (OSM). Having understood the potential and drawbacks of the site, using the strategic planning tool for swot analysis, the project attention was drawn on urban mobility, especially on the traffic decongestion and emission reduction. The mobility hub which is easy to reach from the high transportation demands area of the city and includes various method of transportation was proposed, the first concerning on the mobility hub was the position, taking advantage of macroscopic results, accessibility analysis outcome and an aspect analysis about solar radiation in GIS. Then for automotive area, it was realized a microscopic simulation of intersection with VISSIM, and an emission analysis with MOVES by using the traffic data outcome from the VISSIM simulation, the low poly building batch for the environment and several vehicle models has been prepared by using CityEngine and 3dsMAX. Simultaneously for the civil building engineering area, the setup of BIM model of the parking area has been done.

By using the vehicle models from Sketchup, urban area model from CityEngine and 3dsMax, parking hub model from Revit together and from the microscopic simulation setup in the VISSIM, Zheyuan built the connection between the simulation data from VISSIM and the scene in Unity to make the traffic scene in the digital world. And then based on the various data source such as building information, energy analysis etc. Fanshu built the digital twin traffic data management interface, the traffic scene, BIM model are also integrated in this system. At same time, Zheyuan established the driver first view based interactive traffic scene on the traffic scene and virtual reality (VR) technology. In this thesis, the detailed description of the workflow for the light blue part (**Fig. 3.1**) will be presented.

3.2 Overview of proposal

As mentioned in the section **1.3.2 Structure of thesis and target of case study project** and **3.1 Workflow**, for this project, the construction of a mobility hub and implementation of its digital twin will be proposed. From the result of macroscopic traffic analysis, the swot analysis, and the spatial analysis, the location of the mobility hub is selected as shown in **Fig. 3.2.**

Figure 3.2 Macroscopic traffic analysis from Fanshu

This location has been selected because from the result of macroscopic traffic analysis from TransCAD, traffic and accessibility analysis using GIS technology, population density analysis of the city, this area has a high transportation demand.

The area is positioned at the middle of the Sergio Osmeña Bridge (**Left one in Fig. 3.2**) and Marcelo Fernan Bridge (**Right one in Fig. 3.2**), which means from here both entries of two bridge can be easily reached, it includes the intersection of the M.L. Quezon National Highway which is the one of the highest capacity roads with 3 lanes per one direction and the Lapu-Lapu City Hall road which led to the city hall office of the Lapu-Lapu city. Due to the high-volume traffic of this "T" shape junction, the intersection is equipped with the traffic signal, and there are a lot of Jeepney commuting in this area. Also, many big commercial vehicles usually passing this road segment in order to arrival at the export and industrial region at the Northeast corner of the Mactan-island.

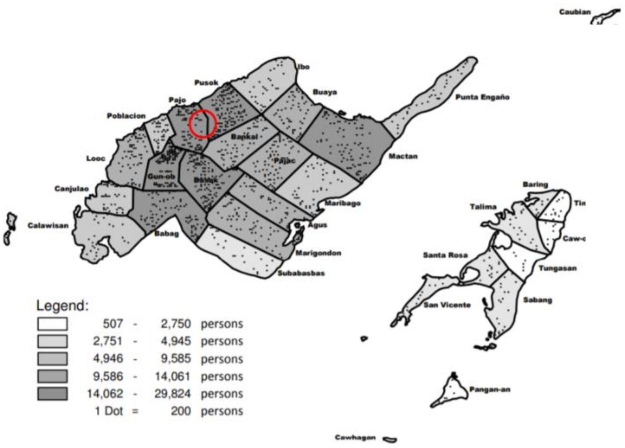


Figure 3.3 Population density by barangays [58]

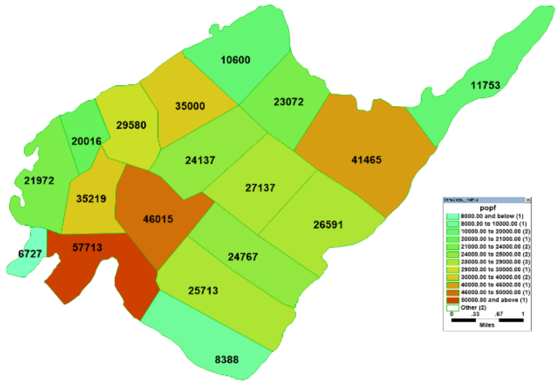


Figure 3.4 Population quantity by barangays from Fanshu

The average population density of Lapu-Lapu city is 7000/km^2, the target area belongs to pajo barangay and near to the pusok barangay, which are 2 area with high population density, 12090/km^2 and 11955/km^2, respectively (In 2015). [59]

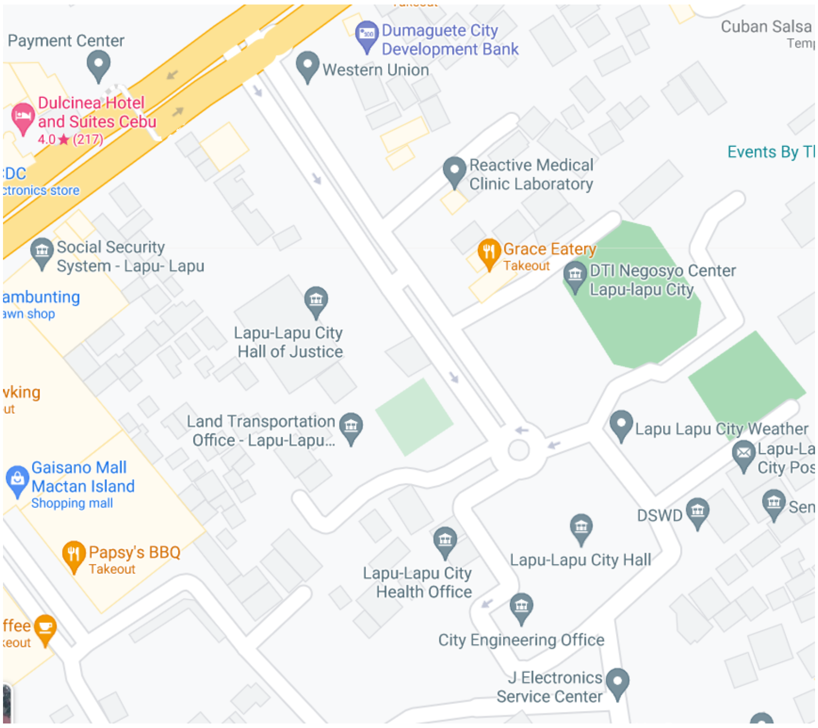


Figure 3.5 Public Services Center

This area is also a resource - intensive area, almost all public service agencies in Lapu-Lapu city are concentrated here such as City Hall, Health Office, Hall of Justice, Land Transportation Office, Police Office etc., the biggest shopping center in the city "Gaisano Mall" is around this area. Such concentrated social resources attract people to this place, the smart mobility hub can increase the fluidity of the area, makes people easily go in and out, avoid the traffic congestion.

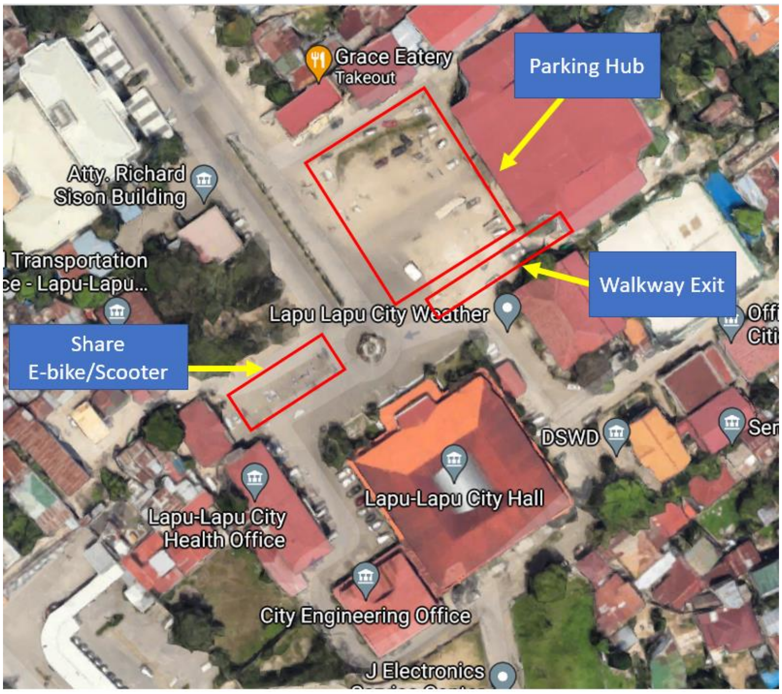


Figure 3.6 Space Configuration of Mobility Hub

The mobility hub includes a smart parking hub, a share electric micro mobility area and the exit of the moving walkway.

The **smart parking hub** is located in front of the DTI Negosyo Center Lapu-Lapu City (an office provides business registration assistance, business advisory services, business information and advocacy, monitoring and evaluation (of business-process improvement for MSMEs)), currently this is an open parking area.

As the development of the electric vehicles, its market share will increase, it is also an important way to combat the environment pollution due to the traffic. Unlike the traditional engine car which can charge the fuel in gas station in 2 minutes and go away, the electric vehicle needs long time to charge its battery, that means the electric vehicles heavily rely on the construction of infrastructure such as the power supply system. At the moment, the insufficient number of electric vehicles charging piles is an important factor that obstruct the development of electric vehicles.

The smart parking hub proposed will provide 34 parking slots for the private car and each parking slot equipped with the charging pile to support the energy supply requirement of the electric vehicle, the various sensor such as camera, ultrasonic sensor, inductive loop etc. will be installed to have various function like automatic charging, free slots detection, safety management, electricity usage monitoring etc. in order to create an safe, electric supported, convenient for disable user parking environment.

Figure 3.7 Moving Walkway Path from Daniele

The **moving walkway** connect the Mactan-Cebu international airport's exit port and the smart mobility hub is proposed (**Fig. 3.7**), with the moving walkway, the path of the walkway will through some important location of the city such as the colleges, the airport, the residential area, the city center etc., makes people travel easily between the different station of moving walkway, and increase the overall transportation capability of surrounding area, greatly alleviated the dilemma caused by insufficient public transportation method. The moving walkway has many advantages such as zero emission, cheap service, speeding up medium-distance travels, good travel panoramic etc.

Considering the narrow roads in the residential area and the area internal of the main roads is not convenient for high volume traffic, the share **micro-mobility** area is proposed, this area located in front of Lapu-Lapu city hall and at left side of Lapu-Lapu city health office as shown in **Fig. 2.19.** Currently, also this space used for an open parking area. The micro-mobility hub will include 40 electric bicycle and 30 electric scooters, the micro-mobility hub will increase the fluidity of the people flow, landing in airport, then coming to the city center through the moving walkway transportation, considering the majority of passenger from the airport does not has the car in the local area, the share e-bike/scooter can be a good choice after exit from the moving walkway.

Instead of charging the micro-mobilities using external charging pile, the battery swap technology will be adopted, which need 2 battery swap station for micro-mobility devices be installed (**Fig.**

2.21), with this method, when the battery finished, just take out it from electric bike/scooter and put in the fully charged battery from the battery swap station in just 1 minutes, this method has been adopted in many cities of China.

Figure 3.8 Battery Swap station for electric bike

The battery swap station can reduce the stress on electricity grid of the city, because without it, during the day time, the high volume of people will continuously using the electric bike/scooter, and the battery consume quickly, and when night coming, the people flow reduced, also it will be the charging time of the electric bike, but this moment is also the peak hour of electricity usage, because the people will use the electricity for lighting, cooking, showing etc. after work and back to home, so to avoid the extra stress on grid during peak hour, battery station has been adopted, it can balance the charging time of the micro-mobility's battery, let the battery charge also during the day time.

As mentioned in **1.1.2 Emission relative to electrical vehicle**, the electric vehicle is not "zero emission" if we consider its production and electricity generation phase, if all the electricity is generated from the fossil energy (coal, oil etc.), the electric vehicle could be worse than traditional engine car in emission point of view. To promote the use of renewable energy, meet the electricity demand of the mobility hub and create a green and pollution-free transportation environment, the solar panel will be installed on the roof of the smart parking hub, the top of the moving walkway corridor, and the shed with solar panel on the top will be used for the micro-mobility hub.

Manufacturer	BYD [22]	Atersa Grupo [23]	SunPower [24]	JinkoSolar [25]
Model	BYD 320P6C-36	A-320P GSE	E19-320	JKM320PP-72-V
Peak power P_m (Wc)	320	320	320	320
Power tolerance (%)	0 - 5%	±1.5%	+5/−0%	±3%
Maximum power voltage V_{mp} (V)	36.78	37	54.7	37.4
Maximum power current I_{mp} (A)	8.7	8.65	5.86	8.56
Open circuit voltage V_{oc} (V)	46.39	45.5	64.8	46.4
Short circuit current I_{sc} (A)	9.15	9.17	6.24	9.05
Module efficiency (%)	16.5	16.49	19.8	16.49
Temperature coefficients of $V_{oc}\mu_{V_{oc}}$ (%/°C)	−0.31	−0.33	−0.176	−0.30
Temperature coefficients of $P_m\mu_{P_m}$ (%/°C)	−0.39	−0.43	−0.38	−0.40
Temperature coefficients of $I_{sc}\mu_{I_{sc}}$ (%/°C)	0.07	0.05	0.035	0.06
NOCT (Nominal operating cell temperature) (°C)	45 ± 2	45 ± 2	45 ± 2	45 ± 2
Number of cells	72	72	96	72
Area (m²)	1.94	1.94	1.63	1.94

Figure 3.9 Electrical characteristics of photovoltaic modules [60]

By using the solar panel, A-320P GSE (**Fig. 3.9**) which has output power 320 walt at peak hour as reference, and from the spatial analysis from Daniele, the space on the roof of the walkway and parking hub is enough for 2650 solar panels, which can output 850 kw in peak hour (good solar radiation).

Type	Capacity	Time to charge	Our recommendation
Regular electricity outlet	2.3 kW	10h30m	This charging method is intended for emergencies only.
EVBox 1-Phase, 16A	3.7 kW	6h30m	This charging station takes longer to fully charge this car.
EVBox 1-Phase, 32A	7.4 kW	3h45m	**This charging station is the best fit for this car!**

Figure 3.10 Charging Power of Electric Vehicle

Type	Charge power	Time to charge	Charge power	Time to charge
Regular electricity outlet	150w	3h45m	71w	4h43m

Figure 3.11 Charging Power of Electric Bike (Left) and Electric Scooter (Right)

Type	Charge power	Time to charge
Regular electricity outlet	30Kw	15min

Figure 3.12 Charging Power of Electric Bus (E-Jeepney)

Concerning the energy requirement by smart mobility hub in busiest condition, which has 34 electric vehicle and 4 electric buses in parking hub, 40 battery of electric bike and 30 batteries of electric scooter in battery swap station charging simultaneously. Reference to the data from the **Fig. 3.10-3.12**, the total power of the solar panel in peak hour is sufficiently meet the energy requirement of the smart mobility hub in worst case (**Fig. 3.13**).

	Number	Power/unit	Total power
Solar panel	2650	320w	850000w

	E-Vehicle	E-Bike	E-Scooter	E-Jeepney	Total	Residual
Power needed	251600w	6000w	2130w	120000w	424130w	423870w

Figure 3.13 Power Analysis of Mobility hub and Solar Panel

Considering the meteorology and geographical information of Philippine, the hour of full sun is 4-5h. In the daytime, the sun radiation level is high, the electricity power generated by the solar panels is more than power needed by mobility hub, but during the night time, the output power from the solar panel is insufficient, to solve this unbalance power supply and demand problem, solar panels should be connected to the grid, and the grid will allocate energy, then during peak hour of the solar panels, the grid can supply the excessive energy generated to residential usage around the walkway or store this energy using Energy Storage Technologies in Smart Grid such as pumped water storage, compressed air energy storage, flywheel energy storage, superconducting magnetic energy storage, super capacitor energy storage and battery energy storage etc. and then reuse them during the peak hour of the smart mobility hub. [61]

As mentioned in **2.2 Digital Twin**, the implementation of digital twin is very useful for the planning, construction, management, maintaining etc. for the smart mobility hub and moving walkway. By using various sensor, we can capture many information, and using this information and digital model of the urban area to build the digital twin, this digital twin can generate many services and function by processing various data, predicting the future development.

For example, by using the sensors in the parking hub, we can know the situation in the parking area such as free slot, parking time, energy consumption, safety issue etc. based on this information, some smart parking function can be generated such as automatic charging service, user parking app service etc. also, this precepted information will be stored in to digital twin database, forms the

historical data of the area, with the expansion of historical data, digital twin can accurately estimate the future state of the parking area using AI technology such as how busy will be tomorrow, which is the power requirement level, how long will this car parking here etc.
From the public data like the time of the flight arrival and departure, the schedule of the college, we can estimate the time of people flow will coming, from the real-time data like the rotational speed of conveyer of the moving walkway, the people quantity inside the walkway corridor, we can know if everything is fine in the walkway, avoid the safety issues, by combining these data, and using some historical data such as the time required for a people from the airport landing to the entry of the walkway, the relationship of people flow peak and the schedule of the college, school, digital twin can estimate the people flow in the future and if it is exceed the capacity of the walkway, if the estimated people number is exceed the capacity, digital twin system can control the gate open time and frequency to control the people number before it is become too large and detected by human. After the people flow peak detected, the digital twin can also detect the state of the charge of the battery of the micro mobility through sensor on the battery and wireless communication technic, if there are many micro mobility with low battery state, digital twin can remind the area manager to swap the battery before there are too many people that cannot find the bike/scooter, avoid the area crowding. Bye using the real-time traffic information perceived from the sensors installed on the traffic signal, road light, under road etc. and historical traffic information, digital twin can predict traffic flow and adjust the traffic signal to avoid the traffic congestion, also, combining the traffic data and road network data, digital twin can generate the smart navigate service to help people and vehicle following the path that led to global optimal. The digital twin is very useful during the planning phase of the project, we can build the digital twin of the project before it be constructed, find the problem by the simulation from the digital twin, avoid the waste of material and time, in the next sections, a methodology to build the traffic scene and driving simulator will be described.

3.3 Microscopic Traffic Simulation

As mentioned in **2.2.4 Digital Twin and Smart Transportation**, one important advantage of digital twin is to let things that require high costs or difficult to achieve in the real world be quickly implemented in the virtual world and simulate it, test its functionality. Find and correct the mistakes with digital twin before the project implementation, then the things in real world can be modified or implemented, this process can greatly reduce the loss of materials, time and money caused by project errors.

In the case of our project, the traffic scene of the location selected will be implemented, this implemented traffic scene has two function, firstly it will be integrated in the data visualization management system implemented by Fanshu, in order to help digital twin to visualize the traffic of the area, evaluate the movability of the parking hub, traffic accident analysis, and the traffic control, secondly it will be used to build the driving scene with the combination of the VR technology, this method allow people to have an experience of driving and parking without the limitation of location even if the parking hub not been constructed already, the driving scene also useful for the testing of the automated vehicle's sensor and algorithms in a virtual traffic environment based on real traffic data and road network data.
Microscopic traffic simulation is a way to simulate the operation rules of the single vehicle with seconds or minutes as the time scale, to evaluate the respective and mutual effects of highways, urban road systems, and pedestrian systems. It has the advantages that traditional mathematical models cannot match.

3.3.1 Microscopic traffic simulation behavior model

To establish a simulation system that reflects the real situation, there must be a simulation model that matches it. The established model should be convenient for the simulation system to realistically simulate various actual traffic behaviors in the road network, such as car-following behavior, lane change behavior, overtaking behavior, and reaction behavior to traffic lights at road intersections. [62]

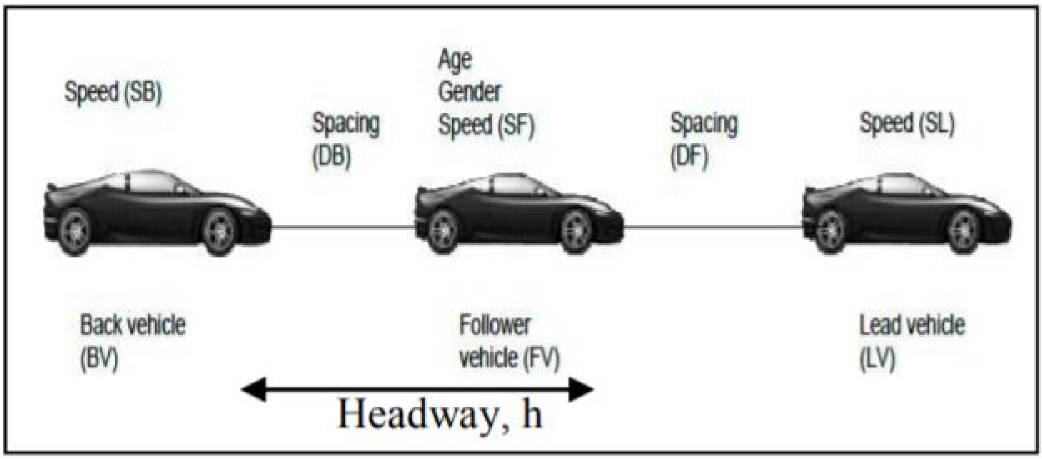


Figure 3.14 Car Following Model from [64]

Car-following behavior describe that during the driving process, the movement of the vehicle is affected by the vehicle ahead. On the one hand, the driver wants to drive at the desired speed, and on the other hand, he must maintain a certain safe distance from the vehicle ahead. However, this effect is asymmetric, because the following vehicle cannot have the same influence on the head vehicle. The car-following equation of the model describes the relationship between the speed, acceleration, and relative position of the front and rear cars. Through solving the equation, the speed, acceleration, and relative position of any vehicle at any given time can be determined. Through further calculation, some other parameters of road traffic flow can be obtained, such as flow rate, average speed, density, etc. [63][64]

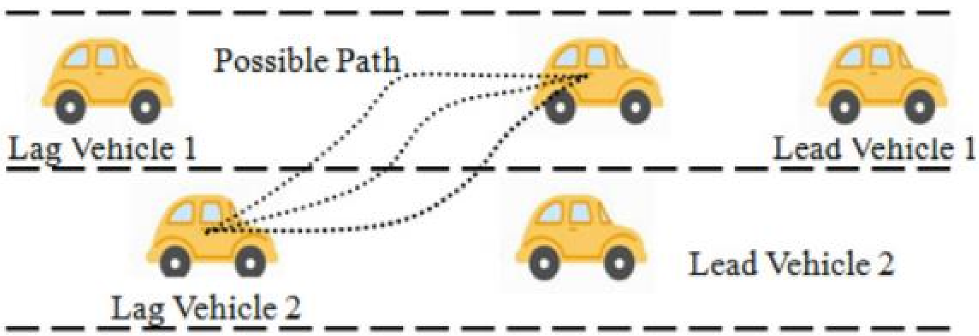


Figure 3.15 Lane Changing Model from [65]

Lane-changing behavior describes the entire behavioral process of a vehicle changing lanes, including the generation of vehicle lane-changing intentions, the feasibility analysis of lane-changing, and the implementation of lane-changing behavior. Specifically, it can be divided into mandatory lane-changing and judging lane-changing. [63][65]

3.3.2 Introduction to microscopic traffic simulation software

With the rapid development of electronic computer technology, the precision and accuracy of computers have been greatly improved. Therefore, microscopic traffic simulation models have begun to appear and are widely used in traffic flow analysis, channelization design, and various signal control schemes optimization and other aspects. In 1976, the traffic simulation software SATURN (Simulation and Assignment of Traffic to Urban Road Networks) was developed by the Traffic Research Institute of the University of Leeds in the United Kingdom to control traffic signals at intersections, it is especially suitable for simulating traffic in small and medium areas. Since the 1980s, due to the rapid development of intelligent transportation system (ITS), many countries have begun to study ITS-related traffic simulation software. So far, a large number of simulation software used to analyze and evaluate ITS have quickly appeared.

At present, the mainstream traffic simulation software includes the microscopic simulation model **CORSIM** (Corridor Simulation), developed by the Federal Highway Administration in the United States, **Paramics** for large-scale road network simulation developed by Quadstone (a company from The United Kingdom), **AIMSUN** for research vehicle navigation systems developed by Spanish TSS, **SUMO** developed by the German DLR Aerospace Center to implement and evaluate researcher's algorithms, multifunctional micro-traffic simulation software **TransModeler** developed by Caliper(a USA company), **VISSIM** for urban traffic control systems and public transportation developed by German PTV (Planungsbuero Transport und Verkehr, Karlsruhe) etc. [66][67]

CORSIM integrates the FRESIM model for microscopic traffic simulation of urban roads and the NETSIM model for highway simulation, forming a comprehensive traffic simulation system that can simulate different traffic scenarios. It divides the entire transportation network into several sub-road networks according to different needs, and each sub-road network is connected to the road section by interface points. At the same time, combined with script programming, the simulation file information can be quickly modified.[68]

PARAMICS uses a combination of different road networks to gradually build a whole road network based on a single node connection method, by correctly simulating the influence of traffic control, ramp control, loops and variable speed limit signs, and variable guidance information. Therefore, it is more complete than the general crowded road network and the conventional simulation software with ITS function. At the same time, the main feature of PARAMICS is internal parallel computing feature, which gives it the powerful ability to simulate large-scale road networks. In addition, PARAMICS also provides API application program interfaces for users to develop personalized operations for different local road conditions and usage needs. [63]

AIMSUN is the traffic flow simulation software developed by the Spanish TSS company. Its initial abbreviation is "Advanced Urban Traffic and Rural Traffic Interactive Simulation System". It includes simulations at various levels from macro to micro, which can be used to integrate transportation planning and Microscopic traffic simulation. The car-following model adopted by the micro AIMSUN is mainly based on Gipps' 1981 safety distance model. [69]

SUMO (Simulation of Urban Mobility) is an open source, micro and multi-modal traffic simulation software. It allows to simulate vehicles moving through a given road network and a given traffic demand. Its simulation allows to solve a large road network's traffic, and it is purely microscopic, that means each vehicle is clearly modeled, has its own route, and passes through the network individually, the randomness of the vehicles is determined by default, but there are different options to change it.

TransModeler is a multifunctional microscopic-traffic simulation software launched by the company Caliper after the transportation planning software TransCAD. It is developed based on the MITSIM simulation software. The car-following model used is mainly the MITSIM car-following model. [70]

VISSIM is a discrete, random, microscopic simulation modeling tool with a time step of one-tenth of a second. It relies on user-defined parameters and uses a psychological-physiological car-following model to reproduce the status of urban signalized intersection traffic and driving behavior. It can analyze the operation status of urban traffic and public transportation under various conditions and is an important tool for evaluating and optimizing the signal timing plan and the optimization plan of the public transportation system in the urban traffic road network. Compared with other simulation technologies, the advantage of VISSIM lies in its detailed and accurate description of traffic facilities (road network, vehicle attributes, signal control, dedicated lanes, bus stops, etc.), which can more truly and accurately reflect the road traffic situation. It can generate online traffic operating conditions for researchers to analyze intuitively and clearly, but also output various statistical data in the form of Excel tables offline to facilitate research and analysis by researchers. Because VISSIM simulates meticulously and needs to input finer background data, the calculation speed is limited by the computer memory and graphics card function adjustment, so to do a large-scale traffic simulation, it needs higher hardware support. [71][72]

In our project, the VISSIM has been adopted to generate/simulate the traffic flow data, not only because of the advantages above mentioned, but also because it provides to user the interface to accomplish the communication between VISSIM and other software (Unity in our case), which makes the easy construction of driving simulator and traffic scene integrated in digital twin system.

3.3.3 Traffic data collection

An accurate source of traffic data is the basis of building a reliable microscopic traffic simulation system. Traffic data include the road network structure, the road type of each link, the traffic signal position and time interval, the traffic flow number, the percentage of each type of vehicle etc. these data can be collected through the manual or automatic way. Among them, manual methods include measuring tape, sighting investigation, and taking pictures on site etc. [73] these method takes long time, consuming many energies and resources, realizing the real-time collection and discrimination of traffic information is a key step in building an intelligent transportation system (ITS). Traffic

Information Collecting System (TICS) uses traffic information collection devices (sensors, cameras, etc.) installed on roads and vehicles to collection, processing and release dynamic information such as traffic flow, driving speed, control information, road conditions, parking lots, weather, etc., TICS is an important part of the ITS. In our case, the traffic information data are mainly collected from the internet resources released from the local government, citizen, and other organization.

OpenStreetMap (OSM) is a free, open source, editable map service jointly created by the Internet public, it is like Wikipedia in the map field. OpenStreetMap is a bit like Google's Map Maker tool. It uses the collective power and unpaid contributions of the public to improve map-related geographic data. Of course, one big difference between it and Google Maps is that OSM is non-profit, and it feeds data back to the community for reuse in other products and services. OSM's maps are drawn by users based on portable GPS devices, aerial photographs, and other free content, the map images and vector data on the website are all marked with shared creative names. The elements of OSM mainly include three types: Nodes, Ways and Relations. These three primitives constitute the entire map screen. Among them, Nodes defines the position of a point in space; Ways defines a line or area; Relations defines the relationship between elements. [74]

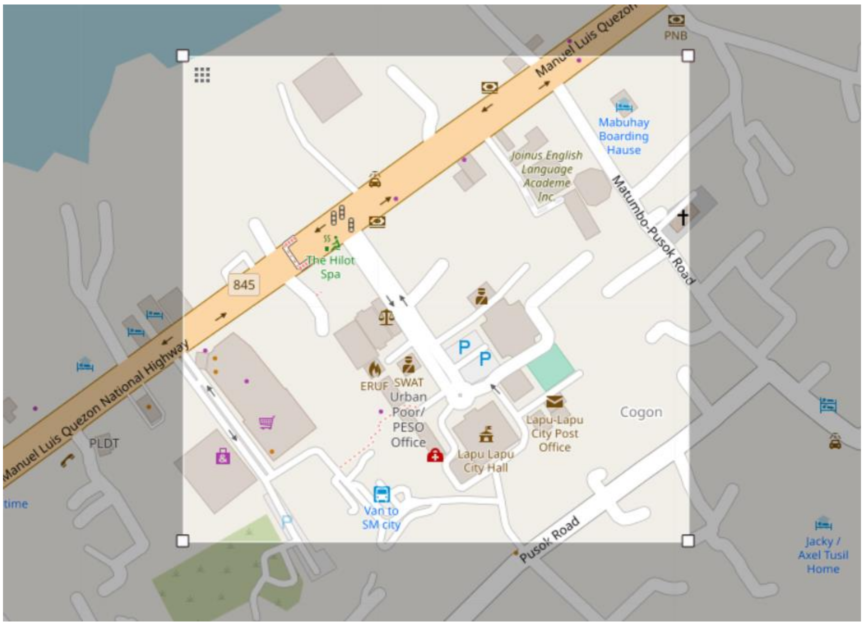


Figure 3.16 Export Map Data from OSM for Area Interested [75]

In the site of OpenStreetMap, we can see a world map, and then zoom to the area interested by us (the Lapu-Lapu city hall area in our case), using the export function showing in the site, we can select the area by edit the highlighted rectangular area in **Fig. 3.16**, then output the map data information file in format **.osm**, this operation process means extract the map data for selected area, and then send these data to the **.osm** file exported. The **.osm** file are coded in XML, Extensible Markup Language (XML) is a structured data exchange language, XML is not a fixed file format, but a language used to define contract formats to facilitate group members to exchange data.

```
 <nd ref="7962019659"/>
 <nd ref="7962019660"/>
 <tag k="highway" v="footway"/>
</way>
<way id="853621719" visible="true" version="1" changeset="91806466"
timestamp="2020-10-01T07:13:37Z" user="GOwin" uid="1041828">
 <nd ref="7962019662"/>
 <nd ref="7962019661"/>
 <tag k="bridge" v="yes"/>
 <tag k="fixme" v="name"/>
 <tag k="highway" v="steps"/>
 <tag k="layer" v="1"/>
</way>
<way id="853621720" visible="true" version="1" changeset="91806466"
timestamp="2020-10-01T07:13:37Z" user="GOwin" uid="1041828">
```

Figure 3.17 Information Included in the .osm file

As shown in **Fig. 3.17**, the information included in **.osm** file is easy to acquire, just need to open it with a text editor, the data in this format is easy to transmit but very hard to understand since it uses pure alphabet and numbers to mark the longitude and latitude, relationship, type etc. information of the nodes and ways, and it has more than one thousand lines even for the little area information exported by us. Hence, convert these data into another format which is easily understandable for people is important.

JOSM (Java OpenStreetMap) is a feature-rich, extensible, and offline OpenStreetMap (OSM) data editor based on the Java language. It supports the GPX track format and can edit OSM information such as waypoints, route lines, metadata, and element relationships. JSOM contains some special-purpose extensions for customers in need to download and install.

Figure 3.18 JOSM interface

With the JOSM, information saved in .osm file can be converted to the nodes and lines distribute in a 2D space which represent the roads, intersections, traffic signals, and buildings. This information now can be easily visualized and edit by the user, in our case, the information of roads and signals of interested area will be used to build the roads network in VISSIM.

As shown in **Fig. 3.18**, the information saved in .osm file are converted into a map formed by line and nodes, the combination of them constructs the elements such as the roads, land use, buildings etc., by clicking the elements, the detailed information on this clicked element will be shown on the right part, the first column “Key” showing the name of different attributes, the second column “Value” showing the values of the attributes corresponding to its name respectively, the number of information depends on how many attributes of this elements are uploaded by the people who edit the map for this area, they can be different element by element, but for the roads, the following attributes usually appears. “highway” specifies the type of the road, street, or path, take the example of **Fig. 3.18**, “primary” means this road is a very important road in the country’s system, which usually used to link the different cities. We should notice that “highway” classifies the roads by them functionality and importance to its country, so even if there are 2 roads both tagged as “primary”, they can be very different if they belong to different country. [76] The attribute “lanes” refers to the number of the lane, “max speed” refers to the speed limit of the road, “oneway” is an attribute identify if the road is one directional or not, in the case of **Fig. 3.18**, the lane’s number are 3, and “oneway” is “yes”, then in VISSIM, we should build this road with 3 vehicle lanes and all of them are only allow the traffic of one direction to pass, if the road is single direction, then on the line which represent this road in the figure map of JOSM, will appear the little arrow to identify which direction it is.

From the data collected on the map and OSM file, we have enough information to build the road network in VISSIM, these data are static data which are unmodified for long time and its very hard to modify or rebuild them.

Another important information needs to be collected and used for the microscopic traffic simulation is the dynamic traffic information such as the traffic flow volume, the percentage of each type of vehicle, the time interval of traffic signal etc. this kind of information is very hard to achieve manually, because it changes hour by hour. In TICS mentioned above, there are different type of sensors can be used to achieve and monitor this kind of information such as inductive loop, microwave detector, video surveillance system, and floating car traffic information collection. The inductive loop is usually composed by a toroidal coil, a transmission feeder, and a detection processing unit. The loop coil is laid under road and forms a magnetic field near it. When a vehicle enters this magnetic field, the detection processing unit detects the vehicle and outputs the signal. It is not just useful for count and detect traffic flow, but also for measure the vehicle speed.

The microwave detector is a type of radar detector that uses the Doppler effect to detect vehicles. It can detect the traffic volume and the vehicle speed, so as to achieve the purpose of detecting road traffic information.

The video surveillance system is mainly composed by front end, transmission, and terminal. Front-end part mainly includes cameras, lenses, pan/tilt, decoders etc. Transmission part includes commonly used optical cables, video cables, telephone lines etc. Terminal part has monitors which can display images from the front-end and can also control the front-end devices.

Floating vehicle traffic information collection is real-time information exchange between vehicles (such as taxis, buses, etc.) installed with global satellite positioning systems (GPS) and wireless communication devices with the traffic data center. It is characterized by a wide collection range, low investment, it can reflect the road network changes in operating conditions and provide a reference for decongestion.

Figure 3.19 Video about the Traffic of the City Hall Junction [77]

In our case, due to the inconvenience of location and the lack of sensors, both manual way and the automatic way cannot be used to get this information, but fortunately, some video recording the traffic flow of the area has been released by the local people which allow us to have a basic knowledge of the traffic of the area. [77]

From the videos, the traffic volume, type of the car, and the percentage of each type of car of 2 minutes are observed, the traffic signal's group and the time interval for each group are counted, this information are used in our microscopic traffic simulation and in next sections they will be shown in detail.

3.3.4 Traffic simulation setup

In this section, the workflow of setup microscopic traffic simulation in VISSIM will be shown.

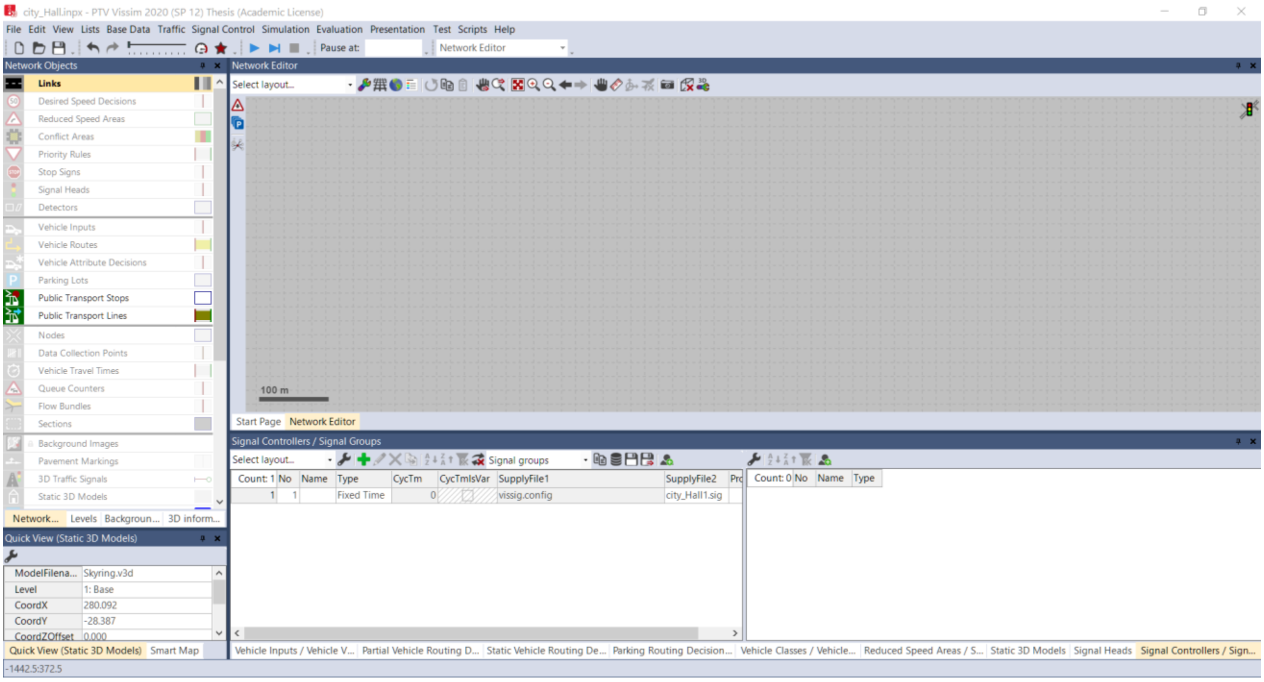


Figure 3.20 VISSIM user interface

The user interface of VISSIM as shown in **Fig. 3.20**, the left section shows to user the various road network object useful for the build of simulation and which object is activated currently, the operation doing in the network editor window is depending on which road network object is activated. The window below the network edit window shows the data list of the objects, it can be used to visualize and edit the data relative to the network objects created. The edit window is the working area, it will show the various objects created and the simulation process.

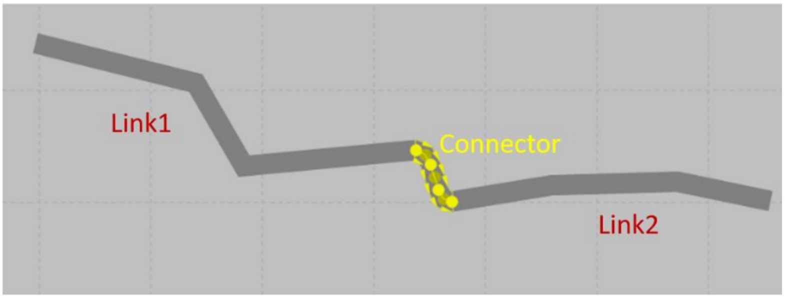


Figure 3.21 Link and Connector

In VISSIM, the road network are composite by link and connector (**Fig. 3.21**), the link is a road segment which the number of lanes, width of each lane, and the shape of the link can be set, the connector can be created only if there are more than one link, because they are used to connect the links and which lanes of one link connect to which lanes of another link correspondingly need to be specified when it is been created. The direction of each link or connector will be defined depend on the direction we draw it, each of them can only has one direction, so if we want to create a road with 2 directions, we must create 2 links with same shape but opposite direction and align them side by side. To create the link, firstly we should activate the "links" in the Network Objects panel (**Fig. 3.20**), then in the "Network Editor" window, the link can be drawn by holding the "Ctrl" and then holding the right button of the mouse, the link segment can be drawn, at the same time, the link with multi-segment can be done with click the left button of the mouse.

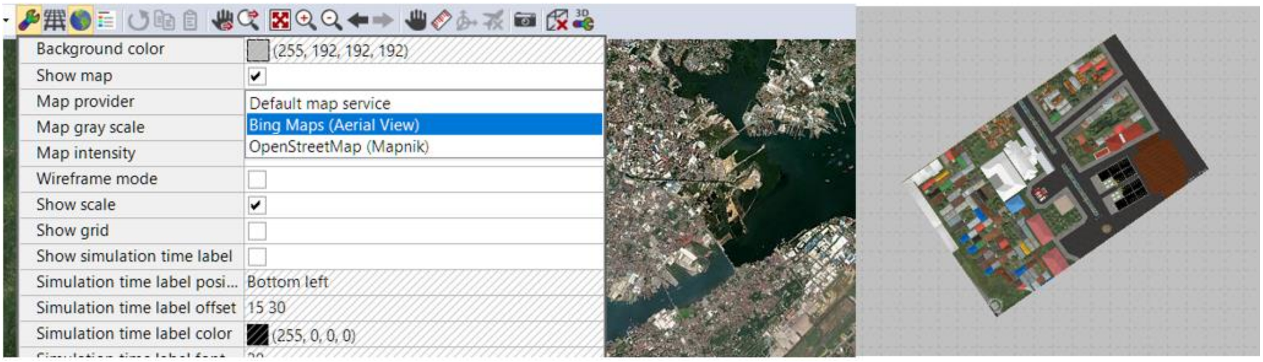


Figure 3.21 Online map and Offline background

When we want to accurately simulate a road network, we need to draw the roads based on the base map, this approach can make the entire simulation more accurate, beautiful, and tidy. In VISSIM, we can use the online map as the background but also, we can import the image of the top view of the area as the reference (**Fig. 3.21**). In our case, the online map has been used for the main roads drawing, and the smart parking hub layout design image been used for the road network in the parking area.

The map background and the top view image can provide the information of the roads position, shape, but there is some information that very hard achieve from the satellite map, this information includes the lane number of the roads, the direction of the lane, the width of the lanes etc. which can be obtained from the osm data file and visualize from the JOSM like mentioned in **3.3.3 Traffic data collection**.

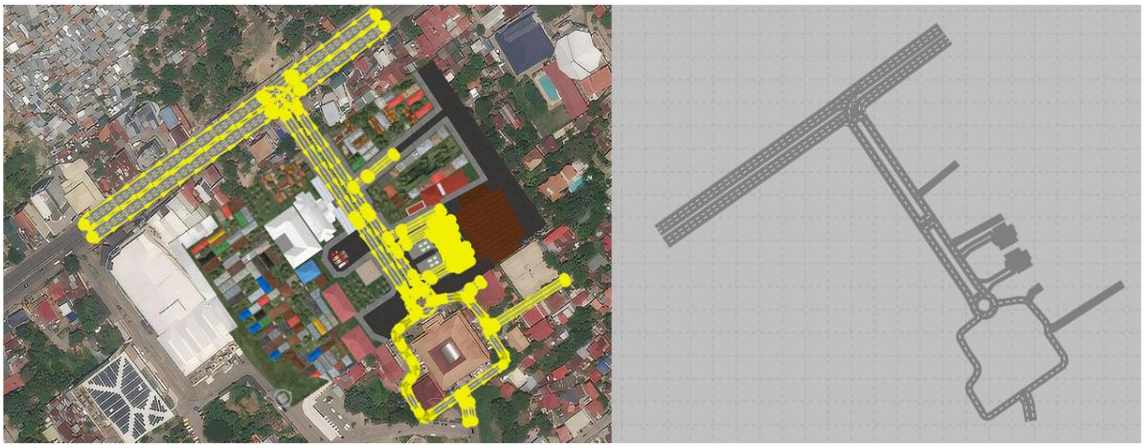


Figure 3.22 Road Network Built in VISSIM

With the roads data from the OSM, the layout aspect from the satellite map, and the design layout of the smart parking hub, the road network has been accurately built, as we can see from **Fig. 3.22**, the road network of the interested area includes a "T" shape intersection, this intersection is formed by the road segment of M.L. Quezon National Highway and the road named Lapu-Lapu City Hall Road which towards to the Lapu-Lapu city hall. The main road segment of M.L. Quezon National Highway is positioned between the entry point of two bridges (**3.2 Overview of proposal**), it has 3 lanes for each road direction, the width of the lane is 3.5 meter, and the two directions of the road are separated by the street green belt, in OSM road classification system, this road been classified as a primary road, which means it is important interconnection between the cities and it has high capacity and high traffic volume. The Lapu-Lapu City Hall road doesn't has the lane information in its osm data, also there are not a clearly lane division lines from the satellite map, but from the traffic video of the intersection and its width, functionally it has 2 lanes for each direction, and again, the two directions roads are separated by the greenbelt and some trees, this road has been tagged as "residential" in osm data, which means this roads serve as an access to housing, without function of connecting settlements, this type of roads often lined with housing, at the end of this road, is the Lapu-Lapu city hall, which is surrounded by a "service" one directional road named Lapu-Lapu City Hall Access Rd, this type of roads are using for access roads to, or within an industrial estate, camp site, business park, car park, alleys, etc. Finally, from the Lapu-Lapu City Hall Road with direction from city hall to the M.L. Quezon National Highway, we can access to our smart parking hub.

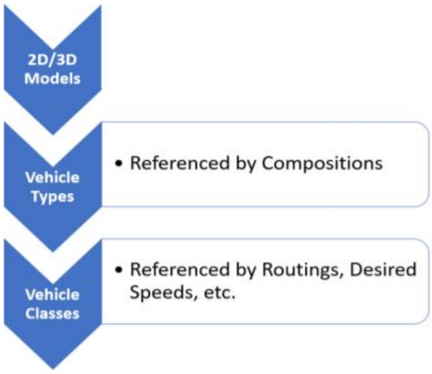


Figure 3.23 Flow of Information Between 2D/3D Models, Vehicle Types, and Vehicle Classes [79]

The combination of the type of the vehicle in the traffic flow is controlled by the setting of 2D/3D Models, Vehicle Types, and Vehicle Classes, VISSIM provides some basic combinations and models by default, but consider the variability and difference of the traffic characteristics country by country and city by city, we are free to add the vehicle model and change the combinations, the 3D model format supported by VISSIM includes ".V3D" by VISSIM, ".SKP" by Sketchup, and ".3DS" by 3ds MAX.

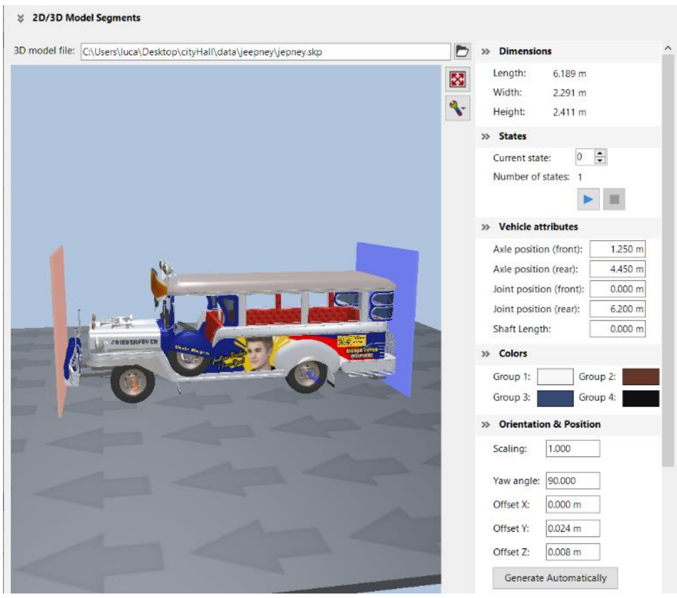


Figure 3.24 Model of Jeepney added into VISSIM

There are many vehicle types in traffic flow, each of them has its own physical property, in which the length and width is very important because its effect the distance between the vehicles, which has effects on the car-following model, over-taking model, and lane-changing model mentioned in **3.3.1 Microscopic traffic simulation behavior model**. The position of the axis of the front and rear wheel has effects on the rotational behavior of the vehicle.

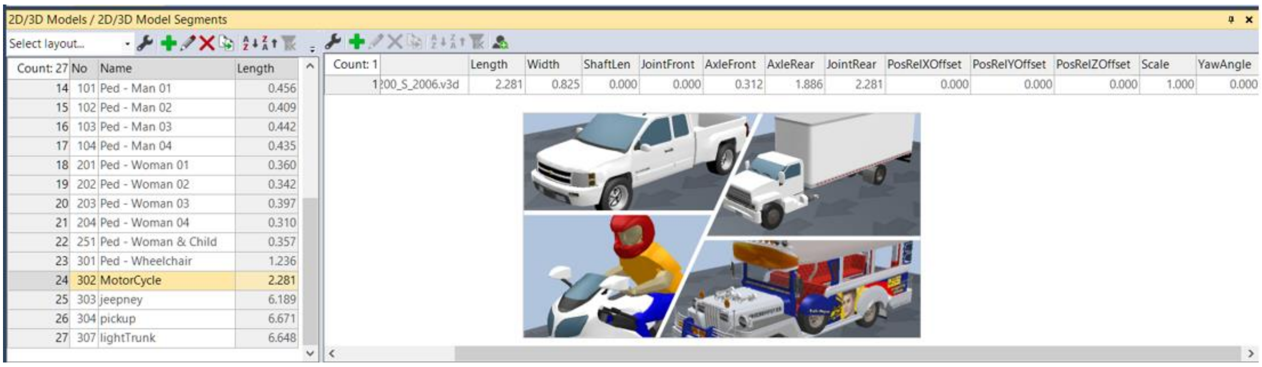


Figure 3.25 List of 3D models

By observing the real traffic video of the intersection [77], the following types of vehicle appeared in the traffic of the intersection: Cars, HGV, Bus, Jeepney, Pickup, Motorcycle, and Light trunk, some of them has already included in the VISSIM's model library, but others (**24-27 in Fig. 3.25**) has to be prepared and calibrate the data by us.

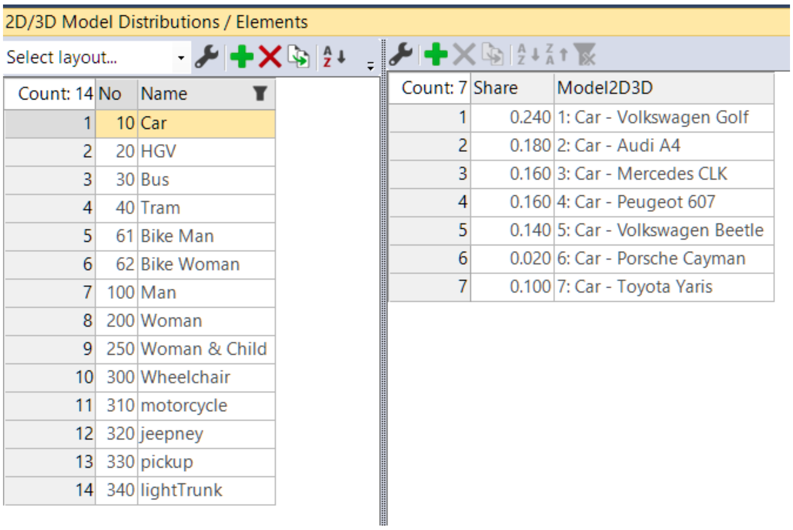


Figure 3.26 Model Distributions for Vehicle

Each vehicle type can corresponding to many different brands which have different models, in the model distribution setting table, the percentage of each model for each vehicle type can be edit, we should pay attention that each vehicle type can only set the models of this type, for example, the models set to car type (**No 10 in Fig. 3.26**) are all the passenger car, if we set a truck model here, then this truck will behave like a passenger car, makes the simulation inaccurate.
There is also the color distribution which can makes the generated car has random color, but since it is not influence on the simulation effect, we kept it as default distribution share.

Vehicle Types

Select layout... <Single List>

Count: 12	No	Name	Category	Model2D3DDistr	ColorDistr1	OccupDistr	Capacity
1	100	Car	Car	10: Car	1: Default	1: Single Occupancy	0
2	200	HGV	HGV	20: HGV	1: Default		0
3	300	Bus	Bus	30: Bus	1: Default	1: Single Occupancy	110
4	400	Tram	Tram	40: Tram	1: Default	1: Single Occupancy	215
5	510	Man	Pedestrian	100: Man	101: Shirt Man		0
6	520	Woman	Pedestrian	200: Woman	201: Shirt Woman		0
7	610	Bike Man	Bike	61: Bike Man	101: Shirt Man		0
8	620	Bike Woman	Bike	62: Bike Woman	201: Shirt Woman		0
9	630	jeepney	Car	320: jeepney	1: Default	1: Single Occupancy	30
10	640	pickup	Car	330: pickup	1: Default	1: Single Occupancy	0
11	650	motorcycle	Bike	310: motorcycle	1: Default	1: Single Occupancy	0
12	660	lightTrunk	Car	340: lightTrunk	1: Default	1: Single Occupancy	0

Figure 3.27 Vehicle Types

In this table (**Fig. 3.27**), the Category (so the operating mode) which determine how the vehicle operating in the traffic, the model distribution, the color distribution etc. can be set to different vehicle types, the capacity column only refers to the public transportation vehicles which need to carry passengers, and need to assign the bus stops for them in order to make it behave like a bus, in our case, there are no bus stop in the interested area, and the jeepney doesn't have the fix stop, which is hard to predict it's stop position, also from the video sources, it rarely stop and let the passengers go on and off at intersection area, so the bus stop behavior is not considered in our case.

Vehicle Compositions / Relative Flows

Select layout...

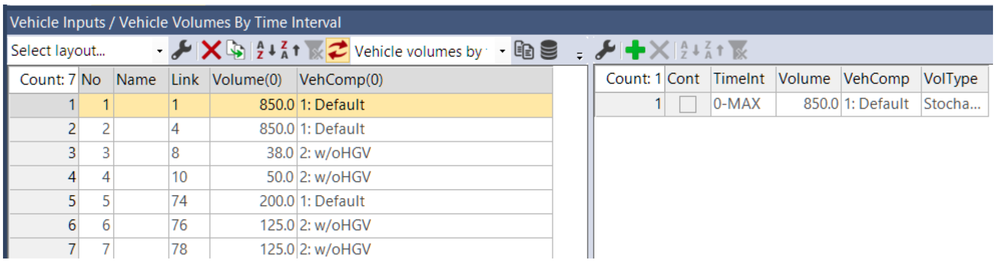
Count: 2	No	Name
1	1	Default
2	2	w/oHGV

Count: 7	VehType	DesSpeedDistr	RelFlow
1	100: Car	50: 50 km/h	30.000
2	200: HGV	40: 40 km/h	2.000
3	300: Bus	30: 30 km/h	1.000
4	630: jeepney	40: 40 km/h	13.000
5	640: pickup	40: 40 km/h	8.000
6	650: motorcycle	60: 60 km/h	41.000
7	660: lightTrunk	30: 30 km/h	8.000

Figure 3.28 Vehicle Compositions

The vehicle types set in **Fig 3.27** are now used to form the vehicle compositions **Fig 3.28**, the desired speed is the speed the car will driving with when there are no other cars or obstructors, the relative flow decide the percentage of each vehicle type for this vehicle composition, the sum of the relative flow don't have to be 1, because the percentage will be calculated automatically by divide its number to the sum of relative flow of all vehicle types. In our case, the number inserted in the relative flow are exactly the average number of each type of car counted in the 2 real traffic videos in [77].

Now we can determine on which points of road networks the vehicle composition should enter and the number of them.



Vehicle Inputs / Vehicle Volumes By Time Interval

Count: 7	No	Name	Link	Volume(0)	VehComp(0)
1	1		1	850.0	1: Default
2	2		4	850.0	1: Default
3	3		8	38.0	2: w/oHGV
4	4		10	50.0	2: w/oHGV
5	5		74	200.0	1: Default
6	6		76	125.0	2: w/oHGV
7	7		78	125.0	2: w/oHGV

Count: 1	Cont	TimeInt	Volume	VehComp	VolType
1		0-MAX	850.0	1: Default	Stocha...

Figure 3.29 Vehicle Inputs

In the vehicle input table (**Fig. 3.29**) we can configure the different input point and its volume, the volume is in number of vehicles per hour, so in our case, the volume in one hour are determined by dividing the vehicle number counted in the videos [77] to the seconds of the video, then multiply them to 3600 seconds (which is the seconds in one hour). For the different time interval, different vehicle input data can be configured, this will makes the simulation more realistic, for example, the traffic volume at peak hour and the traffic volume at normal time phase can be very different, in our case, due to the lack of the field traffic information, the only traffic volume value is set to whole simulation process (which in "TimeInt" column, 850 vehicle per hour for the simulation time 0 to the end for vehicle input point 1).

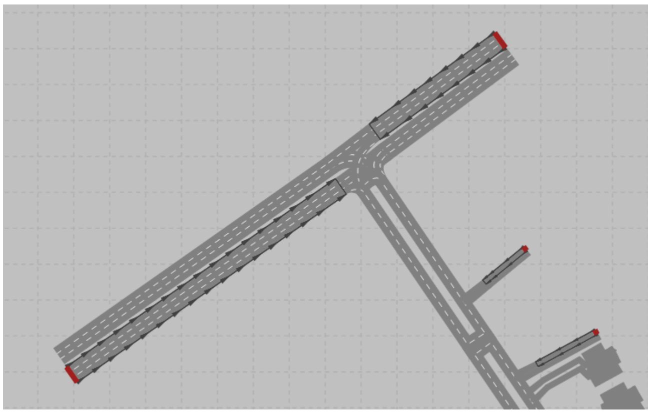


Figure 3.30 Vehicle Inputs Position

The position of vehicle inputs usually is set to the upstream of the links (The red lines at start of the links in **Fig. 3.30**), so the vehicles can enter in the road network from the start of the link, to add the vehicles inputs, firstly we need to activate the “Vehicle Inputs” in the “Network Objects” panel, then click the link which we want add the inputs and right click in the “Network Editor” window, in the menu popped out, select “Add New Vehicle Input”, a vehicle input now has been successfully added to this link, then in the vehicle inputs table (**Fig. 3.29**) we can set the value relative to the vehicle inputs. VISSIM provides two method of the vehicle input, exact and stochastic, where exact will give exactly same traffic flow rate as indicated in the table, stochastic will makes the vehicle input varying around the value inserted in the table according to the random process based on the random seed configured in the simulation parameters.
Now, the vehicle type and the vehicle number has been configured through the vehicle composition and the vehicle inputs, the next step is to define where should the vehicles go and what is the percentage of each route for each vehicle input point.

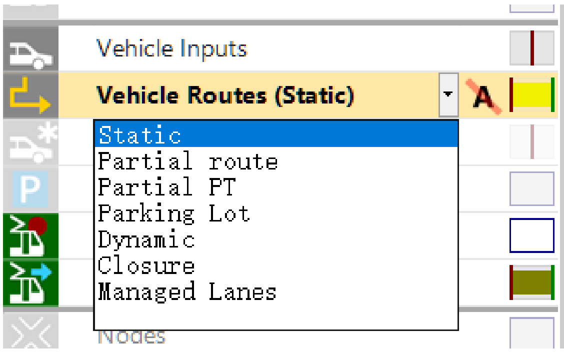


Figure 3.31 Vehicle Route Types

VISSIM provides several vehicles route types to give user the possibility to adapt various traffic aspects (**Fig. 3.31**), the more often used route types are the static route and the partial route. Static route is the basic route type of road network, each vehicle input point should correspond to one static route decision otherwise the vehicles output from this vehicle input point will simply go forward without turning to another way.

Static Vehicle Routing Decisions / Static Vehicle Routes

Count: 7	No	Name	Link	Pos	AllVehTypes	VehClasses	RouteChoiceMeth
1	1		1	7.049	✔	10,20,30,40,50,60,...	Static
2	2		4	5.598	✔		Static
3	3		8	2.327	✔		Static
4	4		10	2.908	✔		Static
5	5		78	3.653	✔		Static
6	6		76	1.958	✔		Static
7	7		74	4.021	✔		Static

Count: 8	VehRoutDec	No	Name	Formula	DestLink	DestPos	RelFlow(0)
1	1	1			3	78.977	0.800
2	1	2			7	26.055	0.015
3	1	7			9	33.064	0.015
4	1	8			73	9.794	0.040
5	1	9			75	64.536	0.020
6	1	10			77	6.939	0.020
7	1	11			3	75.006	0.040
8	1	12			2	173.114	0.050

Figure 3.32 Vehicle Static Routes Table

In our roads network, there are seven vehicle inputs been set to upstream of each link which connected to the roads outside of the interested area, so seven static vehicle routing decision points been set near to each vehicle input point (left table of **Fig. 3.32**). For each routing decision, several routes can be generated by using the routing decision point as the starting point and with the different end point as the target point of each routes.

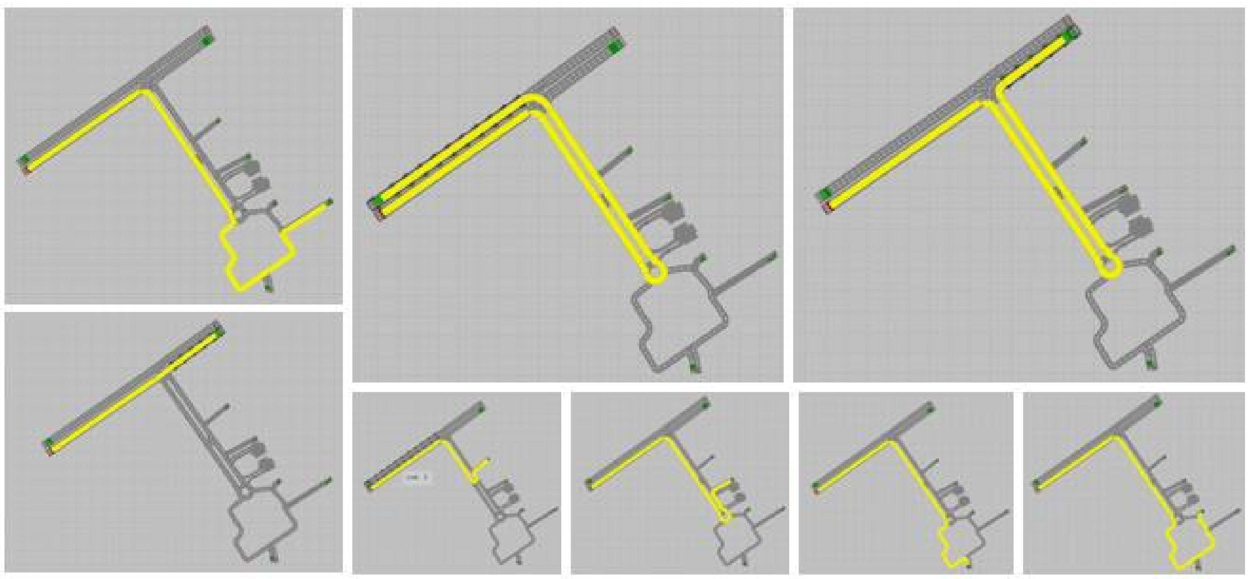
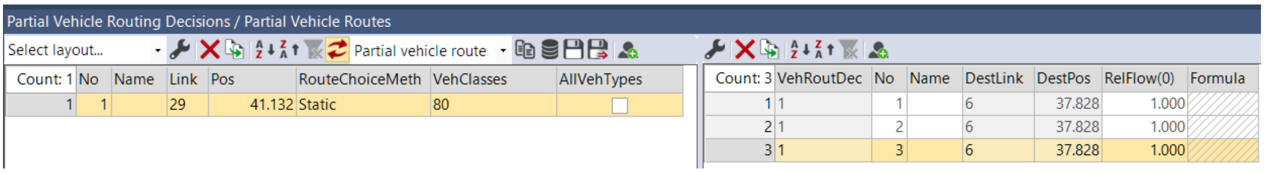


Figure 3.33 Vehicle Static Routes for Vehicle Input point 1

Using the static route decision for vehicle input point 1 (**Fig. 3.29**) as example, like shown in **Fig. 3.33**, each route is started from the static route decision for vehicle input point 1 (the upstream of the link, the red point in the figure) and end to the outport of the road network, each possible route has been considered, and its percentage is defined in the "RelFlow" column of the table (**Fig. 3.32**), again, the sum of them hasn't to be one because them will be calculated based on the weighted average regular, these percentages are observed from traffic videos in [77], to add the routes to the routing decision, the Vehicle Routes should be activated in Network Objects panel, then hold down Ctrl and click the right mouse button in Network Editor window, magenta represents the starting point of the path decision, move the mouse to the end point. The path decision area will be highlighted, after moving to a suitable position, click to confirm the end point, the end point is indicated by blue. The same process needs to be done for all the seven routing decisions, which means more than fifty static routes possibility in our routes network.


Partial Vehicle Routing Decisions / Partial Vehicle Routes

Select layout... Partial vehicle route

Count: 1	No	Name	Link	Pos	RouteChoiceMeth	VehClasses	AllVehTypes
1	1		29	41.132	Static	80	

Count: 3	VehRoutDec	No	Name	DestLink	DestPos	RelFlow(0)	Formula
1	1	1		6	37.828	1.000	
2	1	2		6	37.828	1.000	
3	1	3		6	37.828	1.000	

Figure 3.34 Vehicle Partial Routes

Vehicle partial routes is the secondary routing system of VISSIM, it is used for multiple routing possibility given the origin and destination, that means the vehicle partial routes must be included in the vehicle static routes.

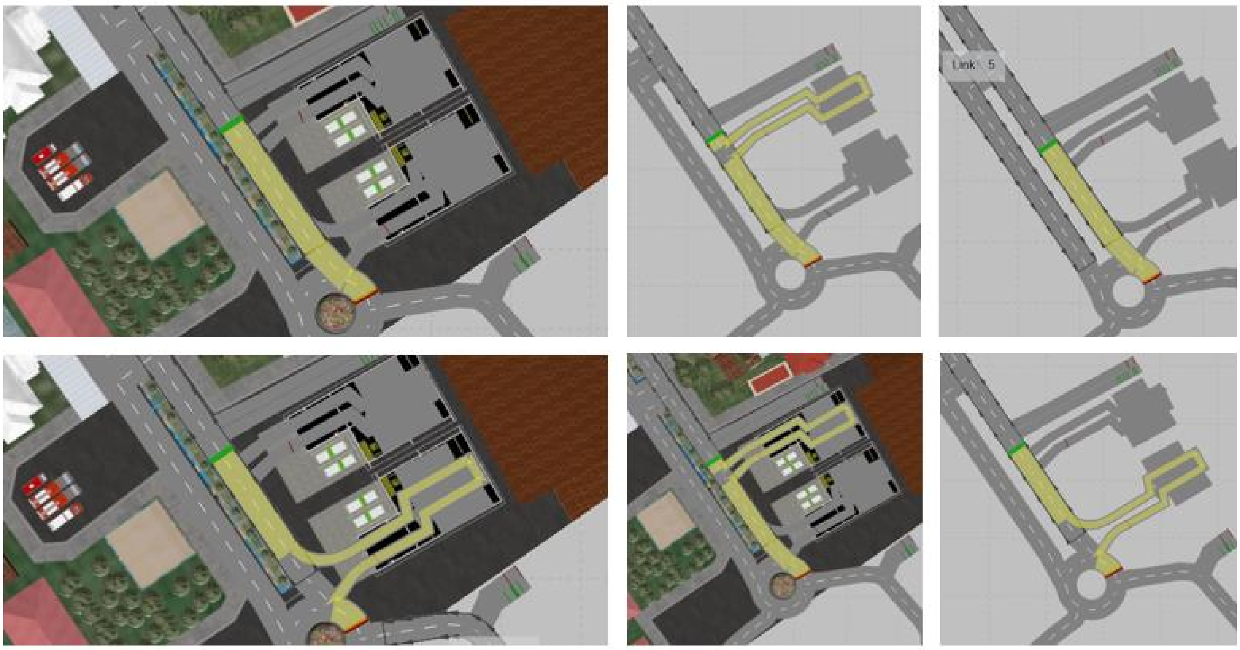
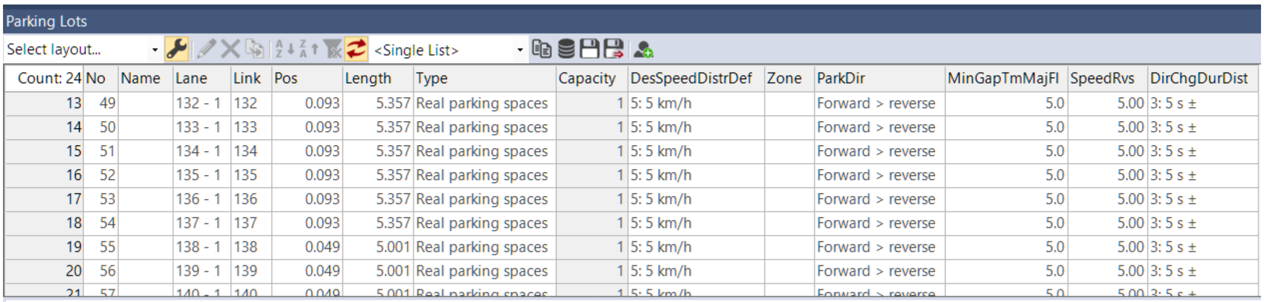


Figure 3.35 Vehicle Partial Routes in Parking Area

The vehicle partial routing decision is suitable for the parking area. The decision flow is following. Vehicle going into the road network from the vehicle input point, when it touches the vehicle static routing decision, the static route will be assigned to this vehicle from the probability shown in the **Fig. 3.32**, then the vehicle will going follow the static route selected, if inside this static route, exists a partial routing decision, once the vehicle touches the partial routing decision (Magenta line in **Fig. 3.35**), the static route assigned to the vehicle is invalid temporary, and a new partial route randomly selected from the routes in the partial routing decision (partial vehicle routes table in **Fig. 3.34**) based on the probability calculated from the relative flow (in partial vehicle routes table in **Fig. 3.34**) will be assigned to the vehicle, then the vehicle follow the partial routes and when it touches the end point of this partial route (Green line in **Fig. 3.35**), the partial routes assigned is finished, the assigned route will become the static route which assigned to vehicle before, the vehicle is still in this static route anyway because as above mentioned, the partial routes has to be included in the static routes, so when vehicles going out from partial routes, they still has to be in the static routes, then this vehicle will follow the static route assigned before and finish its travel in the road network. In our case, there are two parking areas, the vehicle passing by will decide by the partial routing decision if it will park at area A or area B or just go ahead without the parking operation.
After making this partial decision there is still one decision has to be made, that is the parking slot decision.


Parking Lots

Select layout... <Single List>

Count: 24	No	Name	Lane	Link	Pos	Length	Type	Capacity	DesSpeedDistrDef	Zone	ParkDir	MinGapTmMajFl	SpeedRvs	DirChgDurDist
13	49		132 - 1	132	0.093	5.357	Real parking spaces	1	5: 5 km/h		Forward > reverse	5.0	5.00	3: 5 s ±
14	50		133 - 1	133	0.093	5.357	Real parking spaces	1	5: 5 km/h		Forward > reverse	5.0	5.00	3: 5 s ±
15	51		134 - 1	134	0.093	5.357	Real parking spaces	1	5: 5 km/h		Forward > reverse	5.0	5.00	3: 5 s ±
16	52		135 - 1	135	0.093	5.357	Real parking spaces	1	5: 5 km/h		Forward > reverse	5.0	5.00	3: 5 s ±
17	53		136 - 1	136	0.093	5.357	Real parking spaces	1	5: 5 km/h		Forward > reverse	5.0	5.00	3: 5 s ±
18	54		137 - 1	137	0.093	5.357	Real parking spaces	1	5: 5 km/h		Forward > reverse	5.0	5.00	3: 5 s ±
19	55		138 - 1	138	0.049	5.001	Real parking spaces	1	5: 5 km/h		Forward > reverse	5.0	5.00	3: 5 s ±
20	56		139 - 1	139	0.049	5.001	Real parking spaces	1	5: 5 km/h		Forward > reverse	5.0	5.00	3: 5 s ±
21	57		140 - 1	140	0.049	5.001	Real parking spaces	1	5: 5 km/h		Forward > reverse	5.0	5.00	3: 5 s ±

Figure 3.36 Parking Lots Setting

Before setting the parking routing decisions, the parking lots have to be set on the road, the relative attributes of the parking lots (**Fig. 3.36**) includes the link and lane where this parking lot belongs to (that means the parking lots have to be put on the lane of the links), the length of the parking lot (which decide if the car can successfully park on this parking lot, if the length of the vehicle is higher than the parking lot, then this vehicle cannot parking even if the partial routing decision let it go to the park area and the parking routing decision let it to park), the type of the parking lot (in our case is the real parking space), the parking lot’s capacity (one car for one lot usually), the speed of the car in the parking area (normally the vehicle slowing down when it get into parking area), the parking direction (decide if the car park horizontally or vertically), minimum gap time major flow (so that a vehicle may leave the parking space), reverse speed (the speed which the vehicle reversely driving to leave the parking lot), and direction change time duration distribution (refers to the standstill duration between the reverse unparking and subsequent forward movement of the vehicle, this time is not fixed, but follow the random Gaussian distribution).

Time Distributions / Data Points

Count: 9	No	Name	Type	LowerBound	UpperBound	StdDev	Mean
2	2	0.9 s	Normal	0.00	0.90	0.00	0.90
3	3	5 s ±	Normal	0.00	15.00	1.00	5.00
4	20	20 s ±	Normal	0.00	40.00	2.00	20.00
5	30	30 s ±	Normal	0.00	130.00	10.00	30.00
6	101	2.0 s	Normal	0.00	2.00	0.00	2.00
7	102	1.5 s	Normal	0.00	1.50	0.00	1.50
8	103	0.6 s	Normal	0.00	0.60	0.00	0.60
9	104	120 s ±	Normal	0.00	1150.00	100.00	150.00

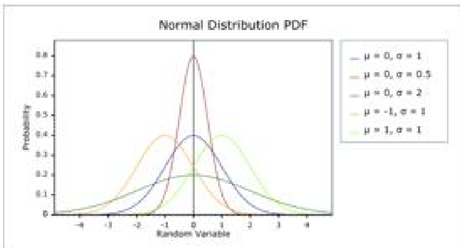


Figure 3.37 Time Distribution and Gaussian Distribution Characteristic

In VISSIM, the time distribution table (**Fig. 3.37**) can configure several time distributions, this time distribution can be used to other functions, such as direction change time duration distribution mentioned above and the parking time which will be used next, the lower bound and upper bound decide the minimum and maximum limit, the type of time distribution usually is the Gaussian (Normal) distribution, the characteristic is shown in the right part of **Fig. 3.37**, mean represent the central axis of the Gaussian curve, the standard deviation effects the flatness of it, more large the standard deviation, more randomly the time around the mean value, if the standard deviation is zero, then this time distribution will exactly equal to the mean value.

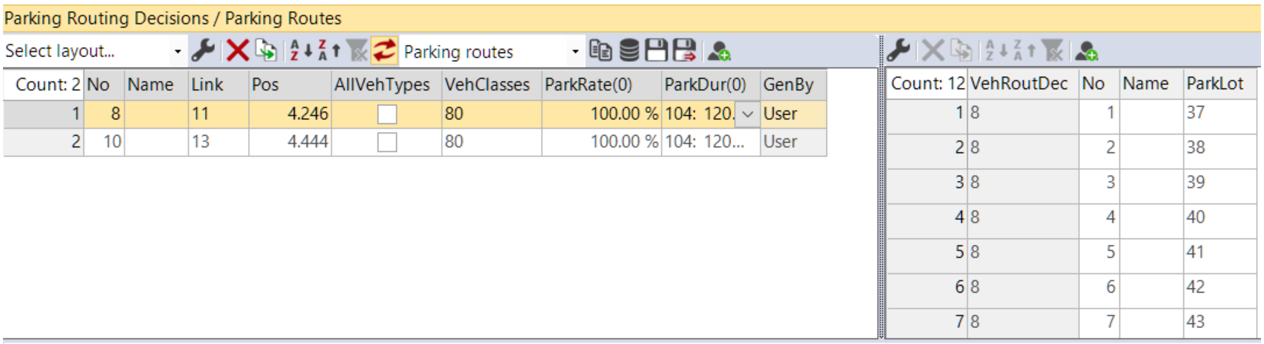
Vehicle Classes / Vehicle Types

Count: 8	No	Name	VehTypes	UseVehTypeColor	Color
1	10	Car	100,630,640,660	✔	(255, 0, 0, 0)
2	20	HGV	200	✔	(255, 0, 0, 0)
3	30	Bus	300	✔	(255, 0, 0, 0)
4	40	Tram	400	✔	(255, 0, 0, 0)
5	50	Pedestrian	510,520	✔	(255, 0, 0, 0)
6	60	Bike	610,620	✔	(255, 0, 0, 0)
7	70	motor	650	✔	(255, 0, 0, 0)
8	80	parkingHub	100	✔	(255, 0, 0, 0)

Count: 4	No	Name	Category	Model2D3DDistr	ColorDistr1	OccupDistr	Capacity
1	100	Car	Car	10: Car	1: Default	1: Single...	0
2	630	jeepn...	Car	320: jeepney	1: Default	1: Single...	30
3	640	pickup	Car	330: pickup	1: Default	1: Single...	0
4	660	lightT...	Car	340: lightTrunk	1: Default	1: Single...	0

Figure 3.38 Vehicle Classes

Several vehicle types (**Fig. 3.27**) can be classified in one vehicle classes by editing the vehicle classes table (**Fig. 3.38**), this classes are useful for the restrict area, for example, some residential areas are forbidden the big commercial truck enter, parking area only for some type of vehicle etc. After all this information have been correctly configured, the parking routing decision can be determined through the parking routing decision table (**Fig. 3.39**).


Parking Routing Decisions / Parking Routes

Select layout... Parking routes

Count: 2	No	Name	Link	Pos	AllVehTypes	VehClasses	ParkRate(0)	ParkDur(0)	GenBy
1	8		11	4.246	☐	80	100.00 %	104: 120.	User
2	10		13	4.444	☐	80	100.00 %	104: 120...	User

Count: 12	VehRoutDec	No	Name	ParkLot
1	8	1		37
2	8	2		38
3	8	3		39
4	8	4		40
5	8	5		41
6	8	6		42
7	8	7		43

Figure 3.39 Parking Routing Decisions

Each parking routing decision assigned with the several parking routes which lead to the parking lots configured in last step, the creating method is similar to the partial routing decision, activating the vehicle routes with the selection of parking lot in the Network Objects panel, create the parking routing decision in Network Editor window by select the link, right click the mouse, and select the "Add New Parking Routing Decision" in the popped-out menu, but in this case, the end point position of the route has to be a parking lot added before. In **Fig. 3.39**, the parking rate define the probability of a parking lot assigned to the car once it contact the parking routing decisions and the parking duration can set the parking time distribution by selecting the time distribution above mentioned.

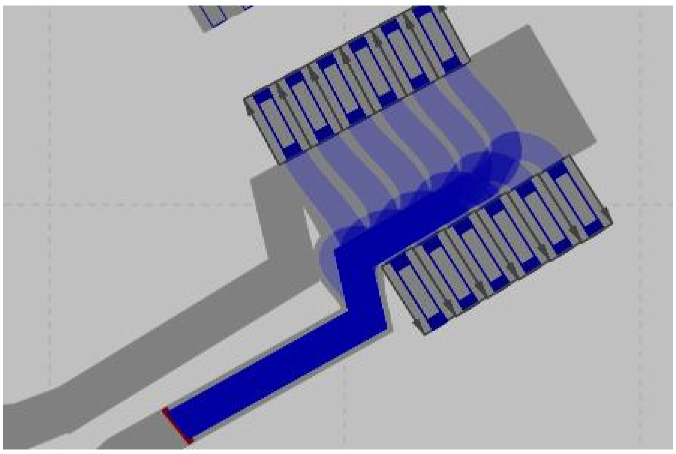


Figure 3.40 Parking Routes and Parking Lots

The route decision process is similar to partial route decision process, vehicle will be assigned a parking lot and a parking time (random time based on the Gaussian distribution) then go to park, after the parking, the vehicle will go out and back to the link which connect the parking lots, the time required for reversing and going out is also follow the Gaussian distribution configured in **Figure 3.36**.

Next, we should take care on the driving behavior which influences the vehicle interaction with each other in the simulation, this kind of behavior can be very different from country to country and can be changed due to the different vehicle composition. In VISSIM, the driving behavior is one of the properties of the link, means that the vehicle interaction mode can be different from link to link, the car following model is mainly Wiedemann 74 and Wiedemann 99, the Wiedemann 74 is suitable for the normal driving behavior in the urban area, the Wiedemann 99 is usually used for the peripherical area or highway driving behavior estimation. In our case, the interested area is located in the city center, so the Wiedemann 74 been selected.

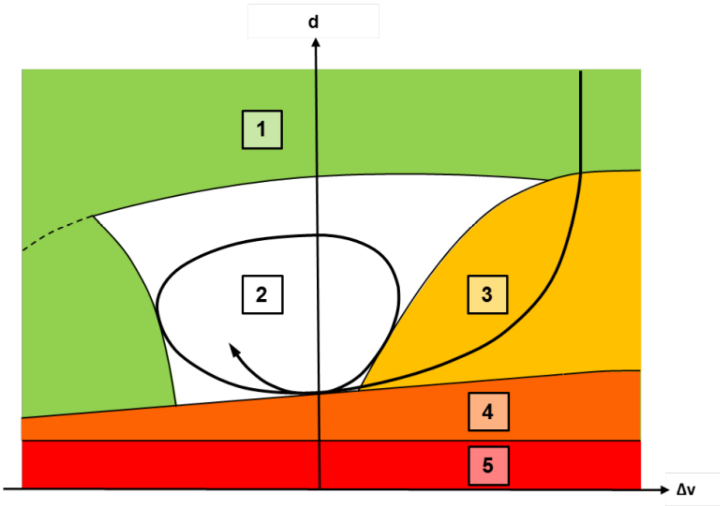


Figure 3.41 Wiedemann 74 Car Following Model [80]

Like shown in **Fig. 3.41**, the vertical axis is the distance between the front boundary of the rear car and the rear boundary of the front car, the horizontal axis is the relative speed between two cars ($V_{Rear} - V_{Front}$), that means when the delta v (horizontal axis) is positive, the speed of rear vehicle is higher than the front vehicle, the distance between two vehicle will decrease, and when delta v is negative, the rear car is slower than the front car, the distance will increase. According to Wiedemann 74 car following model, the driving state can be divided into 5 states, “1” in the green area is "Unregulated behavior" state, where the distance between two vehicles is still high that rear car can drive freely at its desire speed, as decrease the distance between two cars, the driving state enter into "3" in the light orange area, the rear vehicle will realize the existing car in front of him, after a short reaction time (which is different people by people), the rear vehicle will decrease its speed, this state named Approaching state, which represent the rear car is approaching to the front car, then into the "2" white area, in this area, the rear vehicle will behave in following state, it follows the front car by adjust its velocity to keep a safety distance, once the distance is lower than a limit, the rear vehicle will enter into "4" dark orange area, in this area, rear car is in the braking state, which means the driver of rear car realize that they are too close and need to brake to avoid the collision, if for some reason the collision happens, the vehicle is in the "5" red area, the collision state area. [80]

The desired minimum distance of the Wiedemann car following is defined by the equation:

$$\boldsymbol{d} = \boldsymbol{ax} + \boldsymbol{bx} \ (1)$$

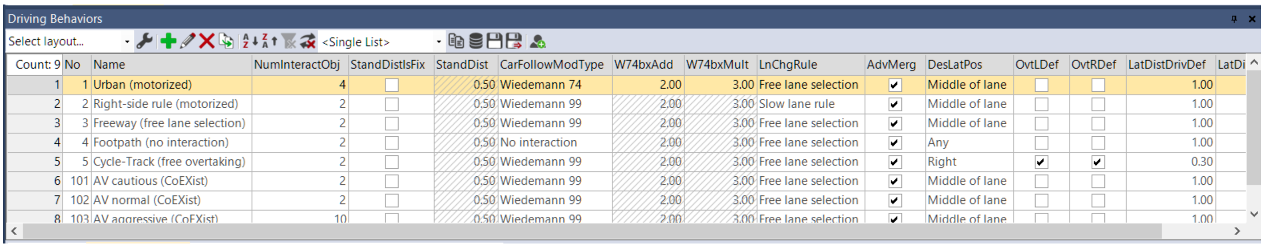
Where $\boldsymbol{ax}$ is the desired distance between stationary vehicles and calculate by equation:

$$\boldsymbol{ax} = \boldsymbol{L_{n-1}} + \boldsymbol{ax_{add}} \ (2)$$

Where $\boldsymbol{L_{n-1}}$ is the length of the front vehicle and $\boldsymbol{ax_{add}}$ is a calibrated parameter.

$$\boldsymbol{bx} = (\boldsymbol{bx_{add}} + \boldsymbol{bx_{mult}} \times \boldsymbol{z}) \times \sqrt{\boldsymbol{v}} \ (3)$$

Where $\boldsymbol{bx_{add}}$ is the additive factor of the safety distance, $\boldsymbol{bx_{mult}}$ is the multiplicative factor of the safety distance, $\boldsymbol{z}$ is a random factor followed by Gaussian distribution with 0.5 as the mean and 0.15 as the standard deviation, $\boldsymbol{v}$ is the vehicle speed in meter per second. [81]


Driving Behaviors

Select layout... <Single List>

Count: 9	No	Name	NumInteractObj	StandDistIsFix	StandDist	CarFollowModType	W74bxAdd	W74bxMult	LnChgRule	AdvMerg	DesLatPos	OvtLDef	OvtRDef	LatDistDrivDef	LatDi
1	1	Urban (motorized)	4	☐	0.50	Wiedemann 74	2.00	3.00	Free lane selection	☑	Middle of lane	☐	☐	1.00	
2	2	Right-side rule (motorized)	2	☐	0.50	Wiedemann 99	2.00	3.00	Slow lane rule	☑	Middle of lane	☐	☐	1.00	
3	3	Freeway (free lane selection)	2	☐	0.50	Wiedemann 99	2.00	3.00	Free lane selection	☑	Middle of lane	☐	☐	1.00	
4	4	Footpath (no interaction)	2	☐	0.50	No interaction	2.00	3.00	Free lane selection	☑	Any	☐	☐	1.00	
5	5	Cycle-Track (free overtaking)	2	☐	0.50	Wiedemann 99	2.00	3.00	Free lane selection	☑	Right	☑	☑	0.30	
6	101	AV cautious (CoEXist)	2	☐	0.50	Wiedemann 99	2.00	3.00	Free lane selection	☑	Middle of lane	☐	☐	1.00	
7	102	AV normal (CoEXist)	2	☐	0.50	Wiedemann 99	2.00	3.00	Free lane selection	☑	Middle of lane	☐	☐	1.00	
8	103	AV aggressive (CoEXist)	10	☐	0.50	Wiedemann 99	2.00	3.00	Free lane selection	☑	Middle of lane	☐	☐	1.00	

Figure 3.42 Driving Behaviors Table

In the driving behaviors table (**Fig. 3.42**), the different driving behavior types can be configured which will be used for the setting of the link property. The attributes of driving behaviors include number of interaction objects (number of preceding vehicles and/or other road network objects such as red signal heads, reduce speed area, stop sign, parking lots etc. near to the vehicle which it will react to), stand still distance (fix stand still distance upstream of static obstacle objects, using the default normal distribution with 0.5 as mean and 0.15 as the standard deviation if this option is not

checked), car following model type (Wiedemann 74 or Wiedemann 99 as above mentioned), additive factor and multiplicative factor which will influence the distance boundaries in Wiedemann model (the factors in equation (3)), lane change rule (free lane selection means vehicle may overtake on each lane), desired lateral position (the desired lateral position in the free flow), overtaken left/right (if is allow the vehicle to overtake others in same lane), lateral minimum distance at 50 km/h and 0 km/h (decide the lateral distance allowed between the vehicles).
To better simulate the vehicle behavior of the area, we need carefully observe the real traffic video of [77].

Figure 3.43 Traffic at the Lapu-Lapu City Hall Junction [77]

From the **Fig. 3.43**, we can see that the vehicles running at the area are quite freely and stay close with each other, when there is a gap between two vehicles, the motorcycle will cut in to go forward, the M.L. Quezon National Highway has higher priority than the Lapu-Lapu City Hall Road, and the vehicle will be slowing down when approaching the traffic signal of the intersection.

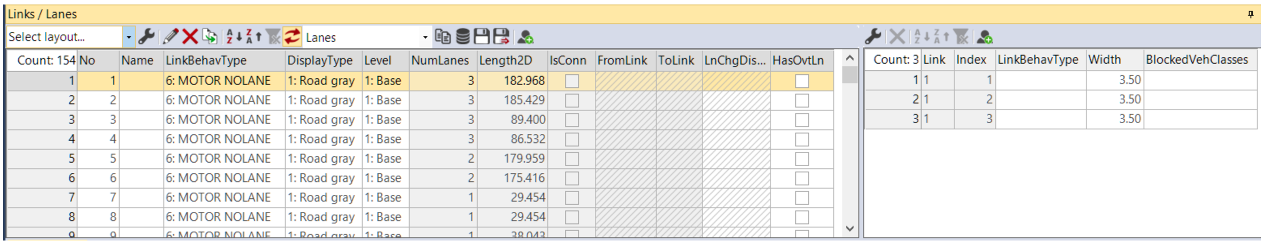
Driving Behaviors

Select layout... <Single List>

Count: 9	No	Name	NumInter...	StandDistIsFix	StandDist	CarFollowModType	W74bxAdd	W74bxMult	LnChgRule	AdvMerg	DesLatPos	OvtLDef	OvtRDef	LatDistDr...	LatDist...
2	2	Right-side rule (motorized)	2	☐	0.50	Wiedemann 99	2.00	3.00	Slow lane rule	☑	Middle of lane	☐	☐	1.00	0.20
3	3	Freeway (free lane selection)	2	☐	0.50	Wiedemann 99	2.00	3.00	Free lane selection	☑	Middle of lane	☐	☐	1.00	0.20
4	4	Footpath (no interaction)	2	☐	0.50	No interaction	2.00	3.00	Free lane selection	☑	Any	☐	☐	1.00	0.20
5	5	Cycle-Track (free overtaking)	2	☐	0.50	Wiedemann 99	2.00	3.00	Free lane selection	☑	Right	☑	☑	0.30	0.10
6	101	AV cautious (CoEXist)	2	☐	0.50	Wiedemann 99	2.00	3.00	Free lane selection	☑	Middle of lane	☐	☐	1.00	0.20
7	102	AV normal (CoEXist)	2	☐	0.50	Wiedemann 99	2.00	3.00	Free lane selection	☑	Middle of lane	☐	☐	1.00	0.20
8	103	AV aggressive (CoEXist)	10	☐	0.50	Wiedemann 99	2.00	3.00	Free lane selection	☑	Middle of lane	☐	☐	1.00	0.20
9	104	NOLANES	4	☐	0.50	Wiedemann 74	2.00	3.00	Free lane selection	☑	Any	☑	☑	0.30	0.10

Figure 3.44 Added Driving Behavior

With the default Urban (motorized) driving behavior in **Fig. 3.42**, the vehicles on road will strictly follow the rules of one row one car on the same lane, even for the motorcycle, they will wait behand even if there are enough space between the car in front of him, which is not happening in reality. to better fit the reality, a new driving behavior has been added (**"NOLANES" in Fig. 3.44**) which makes the vehicle react like the traffic condition in Lapu-Lapu.



Links / Lanes

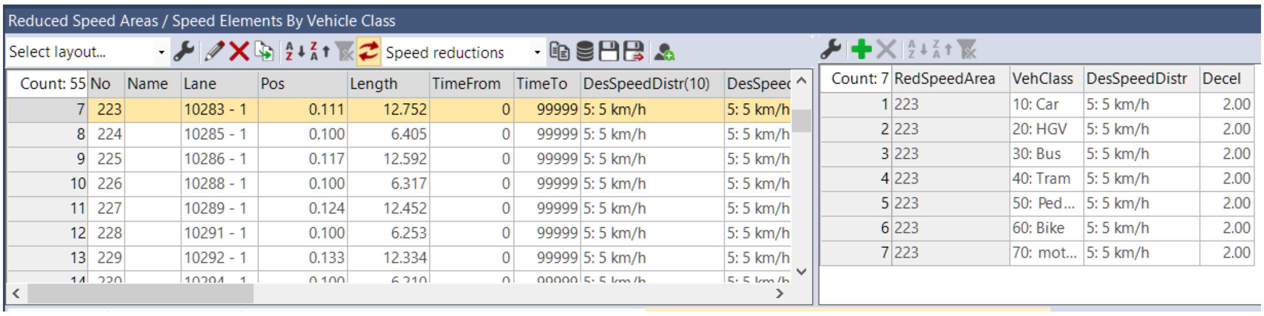
Select layout... | Lanes

Count: 154	No	Name	LinkBehavType	DisplayType	Level	NumLanes	Length2D	IsConn	FromLink	ToLink	LnChgDis...	HasOvtLn
1	1		6: MOTOR NOLANE	1: Road gray	1: Base	3	182.968					
2	2		6: MOTOR NOLANE	1: Road gray	1: Base	3	185.429					
3	3		6: MOTOR NOLANE	1: Road gray	1: Base	3	89.400					
4	4		6: MOTOR NOLANE	1: Road gray	1: Base	3	86.532					
5	5		6: MOTOR NOLANE	1: Road gray	1: Base	2	179.959					
6	6		6: MOTOR NOLANE	1: Road gray	1: Base	2	175.416					
7	7		6: MOTOR NOLANE	1: Road gray	1: Base	1	29.454					
8	8		6: MOTOR NOLANE	1: Road gray	1: Base	1	29.454					

Count: 3	Link	Index	LinkBehavType	Width	BlockedVehClasses
1	1	1		3.50	
2	1	2		3.50	
3	1	3		3.50	

Figure 3.45 Links/Lanes Attributes

In the table of the links and lanes (**Fig. 3.45**), we can configure the relative attributes about links includes the vehicle behavior on this link, the number of lanes, the width of each lane at right part of the table, the existing of the lane for overtaking etc. Also, the connector is a special type of link, the attributes is connected, from link, and to link is the properties for the connector.



Reduced Speed Areas / Speed Elements By Vehicle Class

Select layout... | Speed reductions

Count: 55	No	Name	Lane	Pos	Length	TimeFrom	TimeTo	DesSpeedDistr(10)	DesSpeec
7	223		10283 - 1	0.111	12.752	0	99999	5: 5 km/h	5: 5 km/h
8	224		10285 - 1	0.100	6.405	0	99999	5: 5 km/h	5: 5 km/h
9	225		10286 - 1	0.117	12.592	0	99999	5: 5 km/h	5: 5 km/h
10	226		10288 - 1	0.100	6.317	0	99999	5: 5 km/h	5: 5 km/h
11	227		10289 - 1	0.124	12.452	0	99999	5: 5 km/h	5: 5 km/h
12	228		10291 - 1	0.100	6.253	0	99999	5: 5 km/h	5: 5 km/h
13	229		10292 - 1	0.133	12.334	0	99999	5: 5 km/h	5: 5 km/h

Count: 7	RedSpeedArea	VehClass	DesSpeedDistr	Decel
1	223	10: Car	5: 5 km/h	2.00
2	223	20: HGV	5: 5 km/h	2.00
3	223	30: Bus	5: 5 km/h	2.00
4	223	40: Tram	5: 5 km/h	2.00
5	223	50: Ped...	5: 5 km/h	2.00
6	223	60: Bike	5: 5 km/h	2.00
7	223	70: mot...	5: 5 km/h	2.00

Figure 3.46 Reduced Speed Areas

When vehicle approaching some special areas, its speed will be slowing down, this behavior is achieved by setting the reduced speed areas (**Fig. 3.46**), in its attributes table, we can configure the desired speed and which type of vehicles will reduce its speed in these areas, in our case, the reduced speed area has been set to the parking area and the area near to the intersection.

In some area, there are the conflict between the vehicles from different lanes, for example, the vehicle which want to turn back to another direction and the vehicle which already in this direction and just want to go straight. In our case, the vehicle from the Lapu-Lapu City Hall Road going out and turn left to the west direction of the M.L. Quezon National Highway will conflict with the vehicle that already in this lane and going straight. This situation is inevitable even if we have the traffic signal, in this case, the rule of the priority between the lanes become important.

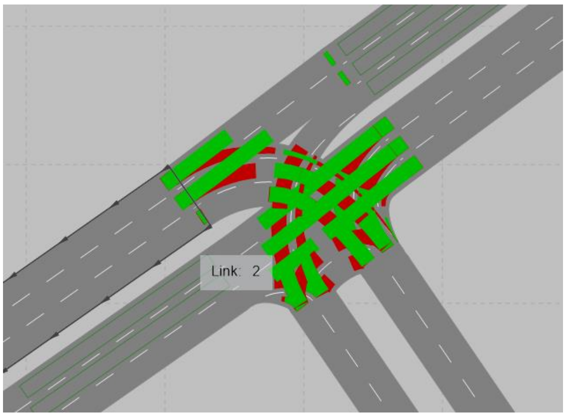


Figure 3.47 Conflict Areas at Intersection

The conflict area can be added by activate the "Conflict Areas" in Network Objects panel and then in the Network Editor Window will appear all the areas conflict between the lanes with the yellow color in the road network, then we need click them to set the priority of them, usually, the primary roads have higher priority than the residential and the turning operation has to make a concession to the straight driving. The light green rectangles behand the intersection are the reduced speed areas above mentioned.

The last step of traffic simulation setup is the traffic signal setting, by observing the intersection videos, the traffic signal time interval has been counted and grouped as **Fig. 3.48**.

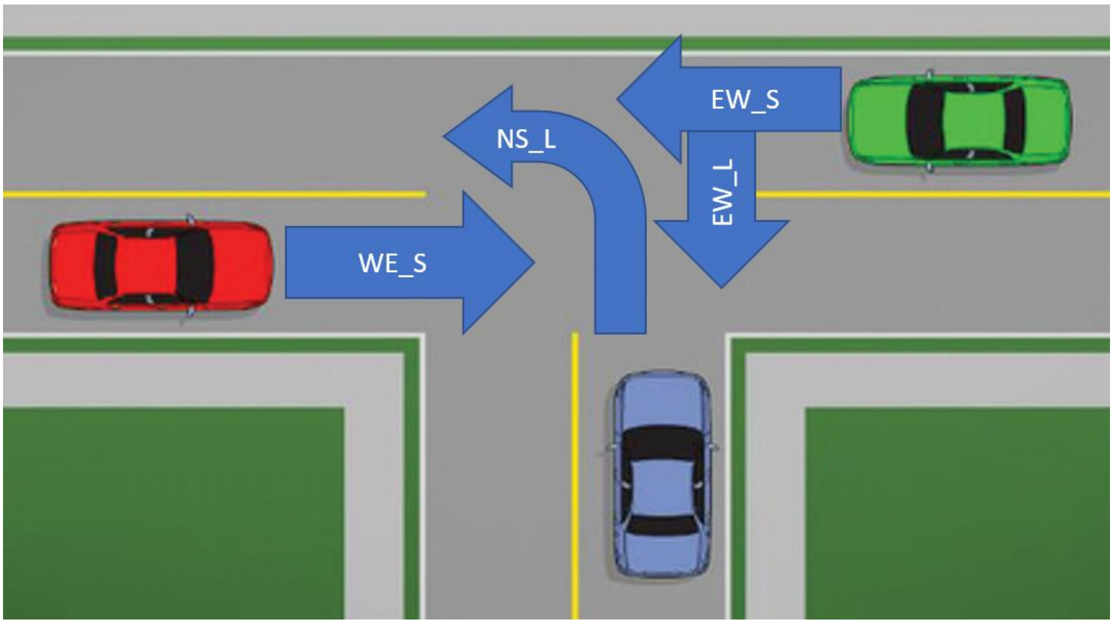


Figure 3.48 Intersection Turning Groups

Where "WE" and "EW" represent two directions of the M.L. Quezon National Highway and "NS" represent the Lapu-Lapu City Hall Road, "L" and "S" means turn left or go straight, the right turning not considered in our case because it does not have to follow the traffic signal, right turn can be done at any time.

The whole traffic signal cycle has 145 seconds, the signal color for each group shown as bellow:

Seconds	Green	Yellow	Red
WE_S	1 to 46	46 to 51	51 to 145
NS_L	51 to 91	91 to 96	96 to 51
EW_L	96 to 141	141 to 1	1 to 96
EW_S	96 to 46	46 to 51	51 to 96

Figure 3.49 Traffic Signal Time Interval

In **Fig. 3.49**, if a time interval is from a bigger number to a small number, means the interval is from the bigger number to the end of current cycle, and then from the start of next cycle to the small number of next cycle, we can also notice that the WE_S, NS_L, and EW_L share the same cycle, which means while one of them is green, the other two must be red, because there are large conflict areas between them operation path. But for EW_S, it has no conflict with the EW_L and no conflict with the WE_S, the only conflict path is NS_L, that means EW_S share the time cycle with the NS_L, when one of them is green, another one has to be red. In other word, the EW_S is red when NS_L is green, which means both WE_S and EW_L are red. By watching the timetable, we can realize that EW_S has the longest green time, which means the traffic stress on EW direction of M.L. Quezon National Highway is lower than the traffic stress on WE direction.

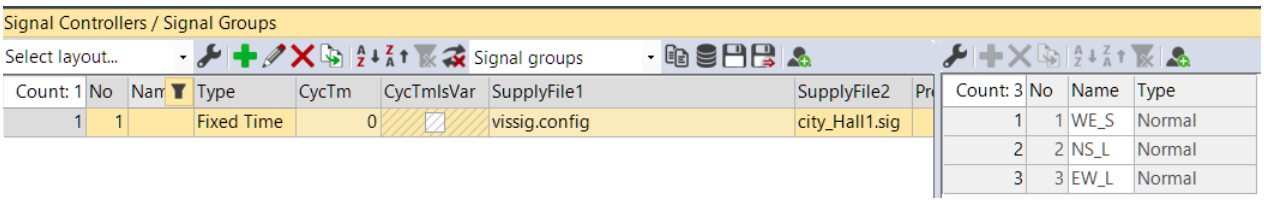


Figure 3.50 Signal Controller

In VISSIM, the traffic signal is managed by signal controllers, a signal controller has to be added before assign the different signal group, in our case, three signal groups should be used, WE_S, NS_L, and EW_L. EW_S haven't been added because it does not share the same cycle time with the others three signal groups, but it can be managed with the bool logic of others signal state.

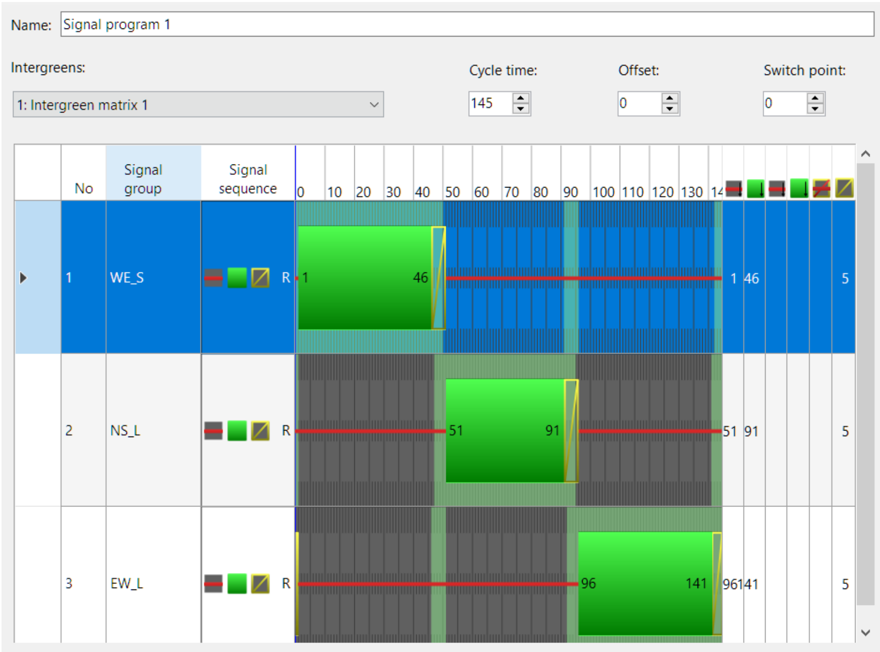


Figure 3.51 Signal Sequence

All signal type is type green, amber, and red; the signal time sequence configured as shown in **Fig. 3.51**. The cycle time is 145 seconds, and the sequence time is configured as above mentioned, the three groups share same time cycle.

Signal Heads

Select layout... <Single List>

Count: 8	No	Name	Lane	Pos	SG	Type
1	1		10188 - 2	0.725	1 - 1: WE_S	Circular
2	3		10188 - 3	0.610	1 - 1: WE_S	Circular
3	4		10189 - 2	1.373	1 - 2: NS_L	Circular
4	5		10189 - 3	1.320	1 - 2: NS_L	Circular
5	6		10192 - 2	1.474	1 - 1: WE_S	Circular
6	7		10192 - 1	1.612	1 - 1: WE_S	Circular
7	8		10191 - 1	1.337	1 - 3: EW_L	Circular
8	9		10188 - 1	0.617	1 - 1: WE_S	Circular

Figure 3.52 Signal Heads

To make traffic lights play a role in traffic simulation, we have to specify on which lane and which position of the lane should have the signal head, then when the signal heads turn to red, the vehicles in simulation will treat the signal head as an obstacle and stop in front of it.

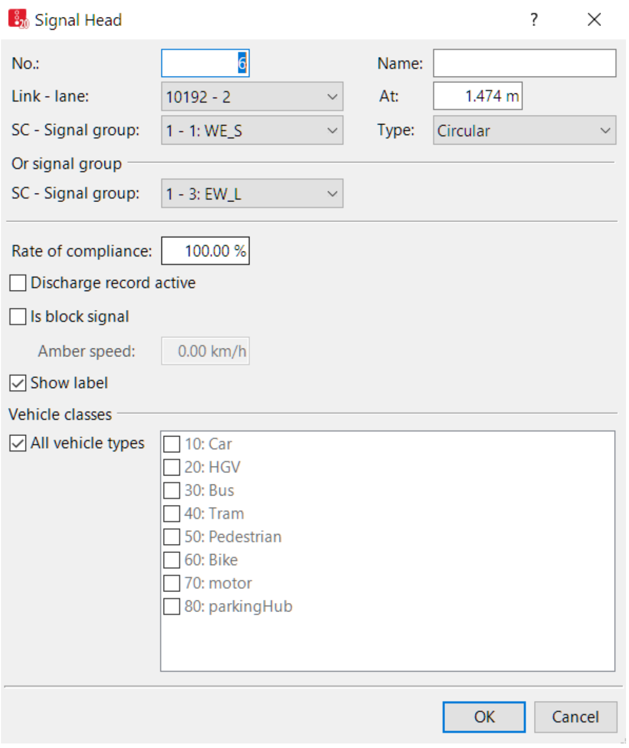


Figure 3.53 Signal Head Setting Panel

In the signal head setting panel, we should assign the signal groups configured above, in order to make it changes the color respect to the signal group time sequence, in our case, as above mentioned, the EW_S signal head can be managed with bool logic with other groups, so the EW_S head is green when WE_S is green or EW_L is green.

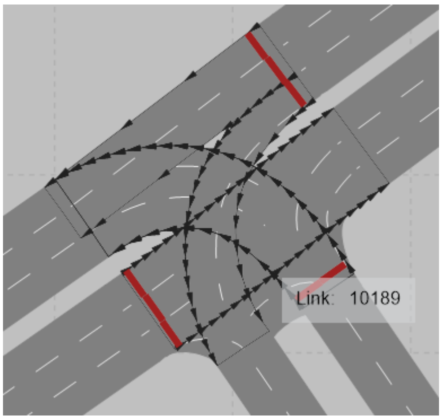


Figure 3.54 Signal Head Position in Road Network

After the traffic signal been configured, the setup phase of the microscopic traffic simulation in VISSIM is finished, the simulation is ready to run at this moment.

3.3.5 Microscopic traffic simulation result

Figure 3.55 Intersection View of Simulation

As shown in **Fig. 3.55**, the traffic simulation has been successfully launched, left one is the 2-dimensional view of the simulation, where the vehicles are represented by blocks with different size and color, the sizes are depending on the configuration in the 2D/3D models table in **Fig. 3.25** and the colors are depending on the color distribution. By press “Ctrl” + “D”, the simulation view is switched to the 3D, and we can see each vehicle models in the scene, as above mentioned, the motorcycle will go forward when there is gap between front vehicles.

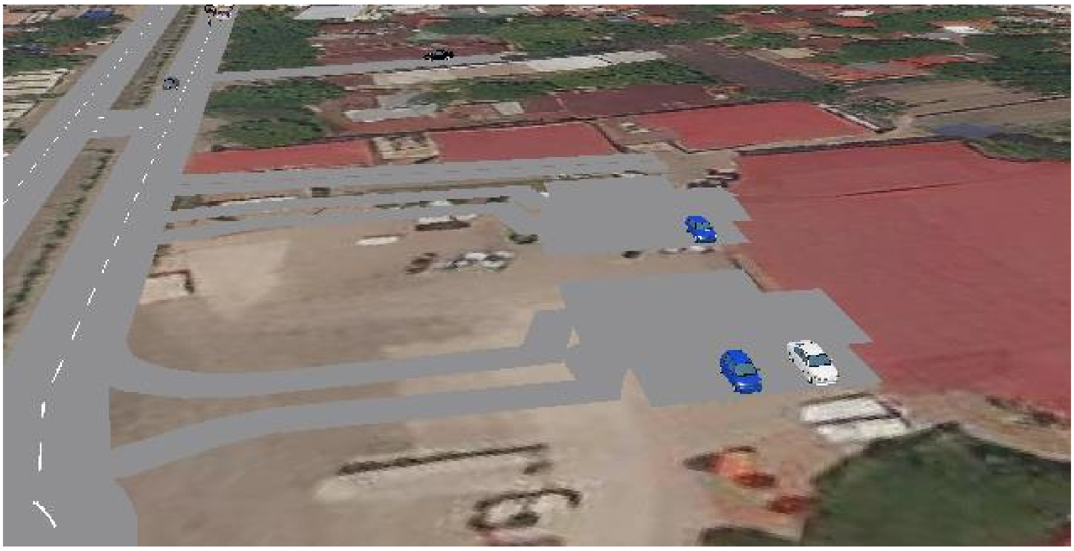
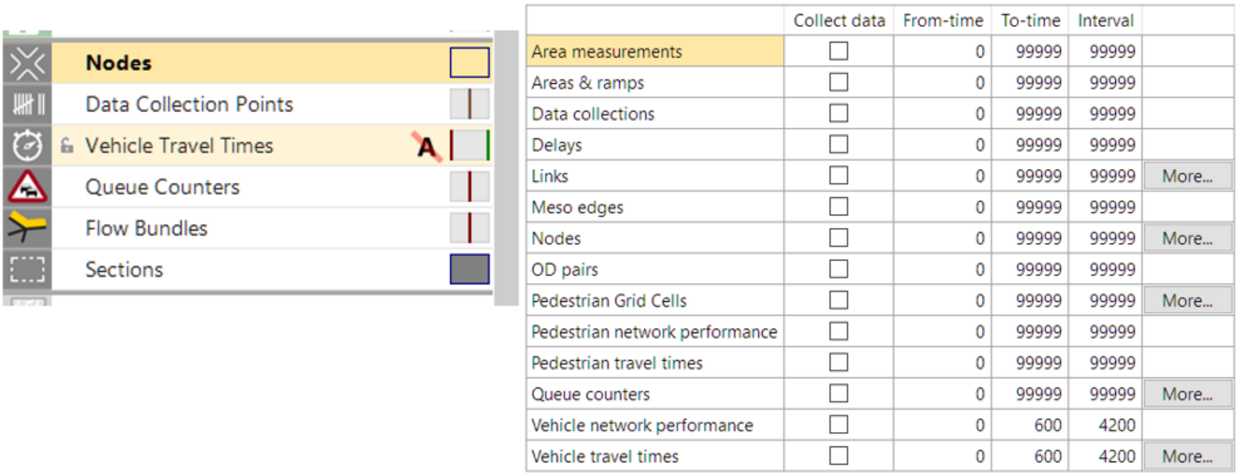


Figure 3.56 Parking View of Simulation

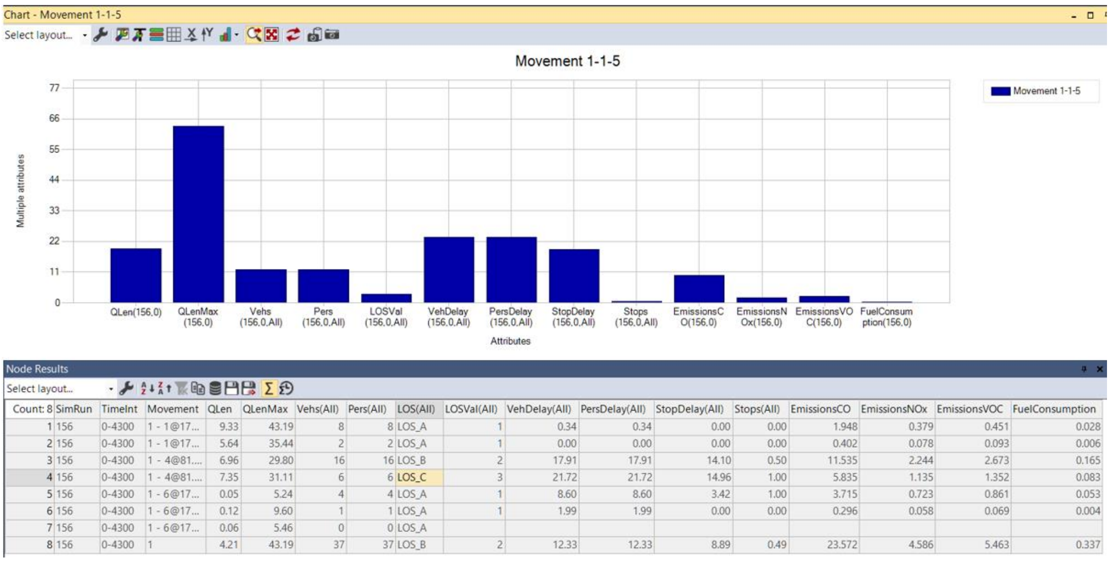
Also, the parking process can be simulated with the propriate configuration of the parking area, the parking lot has configured as only car can enter, so also the partial route decision has to be configured that do not assign the partial routes to the big vehicle like Jeepney and truck.



	Collect data	From-time	To-time	Interval	
Area measurements	☐	0	99999	99999	
Areas & ramps	☐	0	99999	99999	
Data collections	☐	0	99999	99999	
Delays	☐	0	99999	99999	
Links	☐	0	99999	99999	More...
Meso edges	☐	0	99999	99999	
Nodes	☐	0	99999	99999	More...
OD pairs	☐	0	99999	99999	
Pedestrian Grid Cells	☐	0	99999	99999	More...
Pedestrian network performance	☐	0	99999	99999	
Pedestrian travel times	☐	0	99999	99999	
Queue counters	☐	0	99999	99999	More...
Vehicle network performance	☐	0	600	4200	
Vehicle travel times	☐	0	600	4200	More...

Figure 3.57 Data Collection in Simulation

There are several simulation data collection methods to capture the various information from the simulation, which are useful for the road network evaluation. In the evaluation configuration panel (right part of **Fig. 3.57**), we can choose the type of the information we want to know, some of them need to put the detectors (left part of **Fig. 3.57**) in the road network in order to get the information from the specific area, we can decide the start time, the end time, the time interval etc. of the data perception, usually the from-time is not set to zero, because the simulation is started with the empty road network, in this moment the collected data will be lower than the normal data, causing the inaccuracy of the result, so the data collection need a "warm-up" time to let the traffic flow act like in real world and bigger the size of road network, bigger the time required to "warm-up" the simulation.



Count: 8	SimRun	TimeInt	Movement	QLen	QLenMax	Vehs(All)	Pers(All)	LOS(All)	LOSVal(All)	VehDelay(All)	PersDelay(All)	StopDelay(All)	Stops(All)	EmissionsCO	EmissionsNOx	EmissionsVOC	FuelConsumption
1	156	0-4300	1 - 1@17...	9.33	43.19	8	8	LOS_A	1	0.34	0.34	0.00	0.00	1.948	0.379	0.451	0.028
2	156	0-4300	1 - 1@17...	5.64	35.44	2	2	LOS_A	1	0.00	0.00	0.00	0.00	0.402	0.078	0.093	0.006
3	156	0-4300	1 - 4@81....	6.96	29.80	16	16	LOS_B	2	17.91	17.91	14.10	0.50	11.535	2.244	2.673	0.165
4	156	0-4300	1 - 4@81....	7.35	31.11	6	6	LOS_C	3	21.72	21.72	14.96	1.00	5.835	1.135	1.352	0.083
5	156	0-4300	1 - 6@17...	0.05	5.24	4	4	LOS_A	1	8.60	8.60	3.42	1.00	3.715	0.723	0.861	0.053
6	156	0-4300	1 - 6@17...	0.12	9.60	1	1	LOS_A	1	1.99	1.99	0.00	0.00	0.296	0.058	0.069	0.004
7	156	0-4300	1 - 6@17...	0.06	5.46	0	0	LOS_A									
8	156	0-4300	1	4.21	43.19	37	37	LOS_B	2	12.33	12.33	8.89	0.49	23.572	4.586	5.463	0.337

Figure 3.58 Data Collected by Nodes Collector

Fig. 3.58 shows the data collected with the Nodes collector at the intersection area, this information can be visualized in real time during the simulation process, but also can be export to the local file. Among this information, the data about the vehicle dynamic state can be well estimated, because this is what simulation doing, simulate the driving behavior is already includes the data such as the speed, acceleration etc.

In summation, the microscopic traffic simulation building process is followed by **Fig. 3.59**.

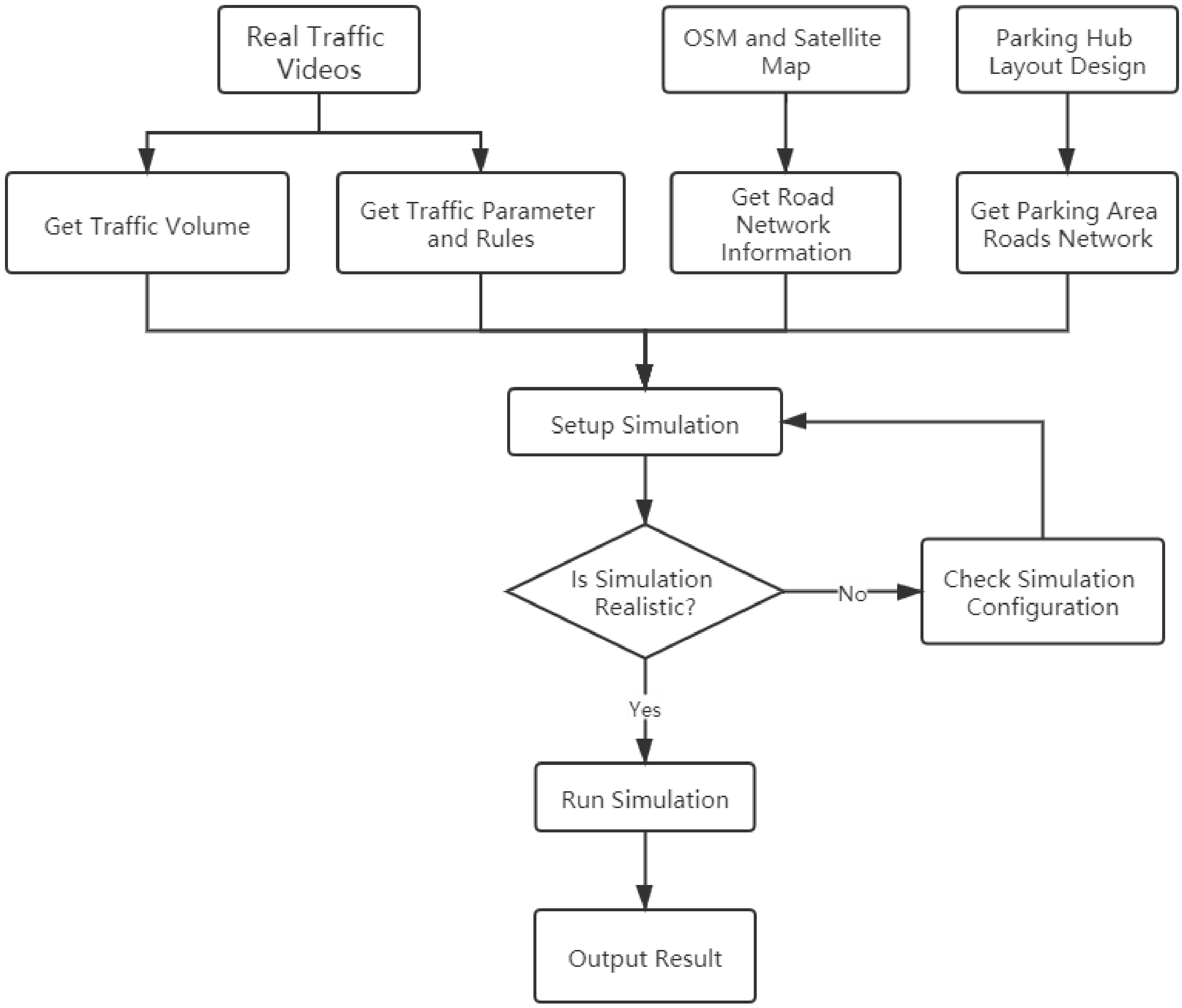


Figure 3.59 Simulation Building Workflow

VISSIM also has the emission estimation function, but there are not so many types of emission substance can be estimate and the evaluation accuracy is not so high, in the next section, the method of using MOVES to estimate the traffic pollution based on the output data from the VISSIM and its result will be roughly described.

3.4 Emission and Energy Analysis Result

As mentioned in the last section, VISSIM can well estimate the data about the vehicle dynamic states such as speed, acceleration, trajectory etc. but for the emission evaluation, the result is not so accurate, to better evaluate the traffic emission in our interested area and how the development of electric vehicle impact on the emission, Fanshu used MOVES to estimate the traffic emission in the interested area for different percentage of electric vehicle based on the vehicle dynamic data output by VISSIM.

3.4.1 Workflow

MOVES (Motor Vehicle Emission Simulator) is a traffic emission estimation software, it was developed by the U.S. Environmental Protection Agency (EPA), the University of California at Riverside, and North Carolina State University, the emission estimation process is based on the MOBILE model and the NONROAD model, which are obtained from a large amount of vehicle test and bench test. Instead of estimating the vehicle emission with the average speed, which is not accurate, MOVES evaluate the emission by considering the operating mode of the vehicle (starting, accelerating, coast down, braking etc.) which is relative to the vehicle specific power (VSP). Also, many experimental studies have proved that vehicle specific power (VSP) can more accurately reflect the relationship between vehicle operating conditions and pollutant emissions than specific speed and acceleration.

The vehicle specific power is calculated from the power divided by mass, where power is depending on many factors include velocity, acceleration, road slop, aerodynamic drag, rolling resistance etc. these factors depending on the vehicle mass, shape, road condition etc. which is very different between different type of vehicle, there are lots of research on these factors and some formula based on the experiment for different type of vehicle was provided in some papers. Through this formula, the vehicle specific power can be calculated knowing the vehicle speed and vehicle acceleration, which can be estimate accurately by VISSIM, therefore, the workflow in **Fig. 3.60** has been adopted to estimate the emission and the energy consumption.

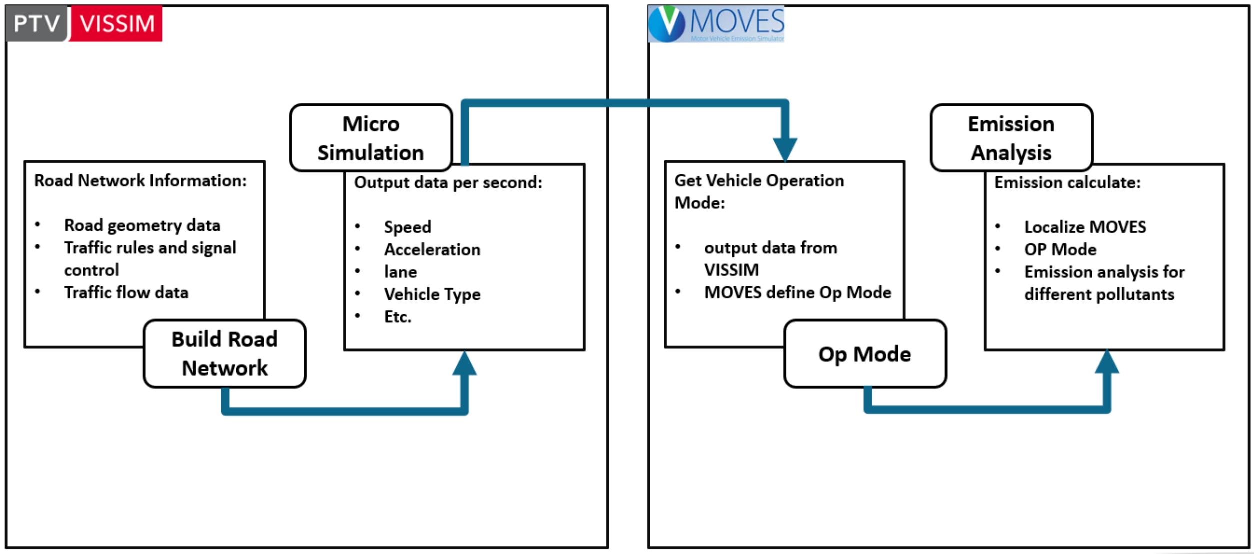


Figure 3.60 Workflow for Pollution Estimation

The several data sources reflect the real traffic condition helps the microscopic traffic simulation building in VISSIM, then run the simulation and output the vehicle dynamic data such as trajectory, velocity, acceleration etc. next, using some algorithm to process the data output from VISSIM and transform it to the data need by MOVES, finally, estimate the emission and consumption by the simulation in MOVES.

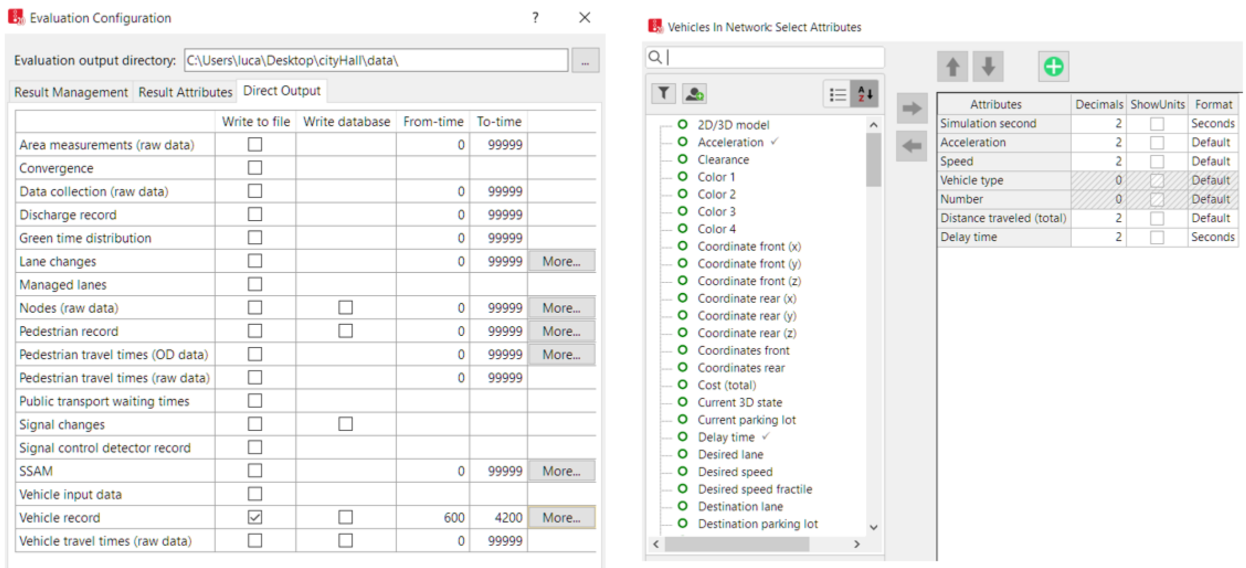


Figure 3.61 VISSIM Output Data Setting for Emission Estimation in MOVES

In VISSIM, we need check the Vehicle Record in evaluation configuration panel, like above mentioned, the traffic simulation needs a time to "warm-up" the road network from empty of start, the desired time in simulation is one hour, so the data output time from 600 second to 4200 second. Then we can choice which attributes do we need by clicking “More” and then “Attribute Selection”, in our case, the data needed by us is shown in right part of the **Fig. 3.61**.

The output data will be saved in a **.fzp** format file, the time resolution was set to 10 step per 1 second, so the information of vehicles will be recorded in this file every 0.1 second, the result is shown in **Fig. 3.62**.

city_Hall_153.fzp

```
$VISION
* File: C:\Users\luca\Dropbox (个人)\aomiao\temp\cebu_vissim\city_Hall.inpx
* Comment:
* Date: 3/10/2021 5:50:45 PM
* PTV Vissim: 2020.00 [12]
*
* Table: Vehicles In Network
*
* SIMSEC: SimSec, Simulation second (Simulation time [s]) [s]
* ACCELERATION: Acceleration, Acceleration (Acceleration during the time step) [m/s2]
* SPEED: Speed, Speed (Speed at the end of the time step) [km/h]
* VEHTYPE: VehType, Vehicle type (Select Vehicle type from the list box)
* NO: No, Number (Unique vehicle number)
* DISTTRAVTOT: DistTravTot, Distance traveled (total) [m]
* DELAYTM: DelayTm, Delay time (Difference between optimal (ideal, theoretical) driving time) [s]
*
* SimSec;Acceleration;Speed;VehType;No;DistTravTot;DelayTm
* Simulation second;Acceleration;Speed;Vehicle type;Number;Distance traveled (total);Delay time
*
$VEHICLE:SIMSEC;ACCELERATION;SPEED;VEHTYPE;NO;DISTTRAVTOT;DELAYTM
600.10;0.00;0.00;630;280;444.24;78.25
600.10;0.00;40.53;630;290;377.42;84.85
600.10;0.00;0.00;630;307;451.64;21.02
600.10;0.00;0.00;630;309;429.58;46.52
600.10;-0.27;62.28;650;317;370.14;87.60
600.10;0.23;37.89;630;322;287.91;74.71
600.10;-0.23;32.68;660;330;283.71;55.61
600.10;-0.20;60.70;650;333;319.08;66.67
600.10;1.44;35.80;650;336;262.50;57.15
600.10;0.27;42.95;640;337;378.53;13.99
600.10;-0.15;38.79;100;339;259.61;48.78
```

GB2312, Line 5, Column 27

Figure 3.62 Output Vehicle Record Data from Traffic Simulation

The output data includes the vehicle records in one hour simulation time, the time requested for simulation is not one hour, because the simulation speed can be changed, so the time in simulation and time in real world is not 1:1, depending on the computer performance, the one hour in simulation time can be finished in some minutes. The data has to be processed using some data processing tool such as Python, Excel, MySQL etc. because there are more than one hundred thousand line of data which is impossible to processing them manually. Then these data have been used for the emission and consumption estimation in MOVES (more detailed description of workflow is included in Fanshu’s part).

3.4.2 Result

Type	Number	Percentage
Motorcycle	937	41%
Car	669	29%
Light Truck and Pick-up	385	17%
Jeepney	248	11%
HGV	38	2%
BUS	22	1%

Figure 3.63 Vehicle Counts in Simulation

First, let us take a look at the vehicle composition during this one-hour time in simulation (**Fig. 3.63**). as we can see, there are more than 2000 vehicle involved during this hour, among them, the motorcycle shares the most percentage, which is fit to the real vehicle share data of Philippine in **Fig. 1.9** and the vehicle percentage observed from the videos [77], the number of commercial vehicle is high because the North East end of M.L. Quezon National Highway is an export area which includes many industrial facility.

Type	Distance [Km]	Total Energy [KJ]	CO_2[g]	CO_2 Percentage
Motorcycle	344.4	1239272.32	89062	0.27
Passenger car	244.6	1291159.482	90940	0.28
Passenger truck	90.1	603272.37	42634	0.13
Light commercial truck	140	1066856.112	75541	0.23
Transit bus	8	131944.138	9507	0.03
Refuse truck	12.9	240759.359	17428	0.05

Figure 3.64 Total Distance and Total Energy

Fig. 3.64 showing the energy use and CO2 emission for each type of vehicle, the Jeepney has been classified as the passenger truck because there are no experiment pollution data for Jeepney, its emission data should be higher than the passenger truck because Jeepney stop and start more frequently (no fix stop), and its average age is elder than the normal age distribution of the vehicle.

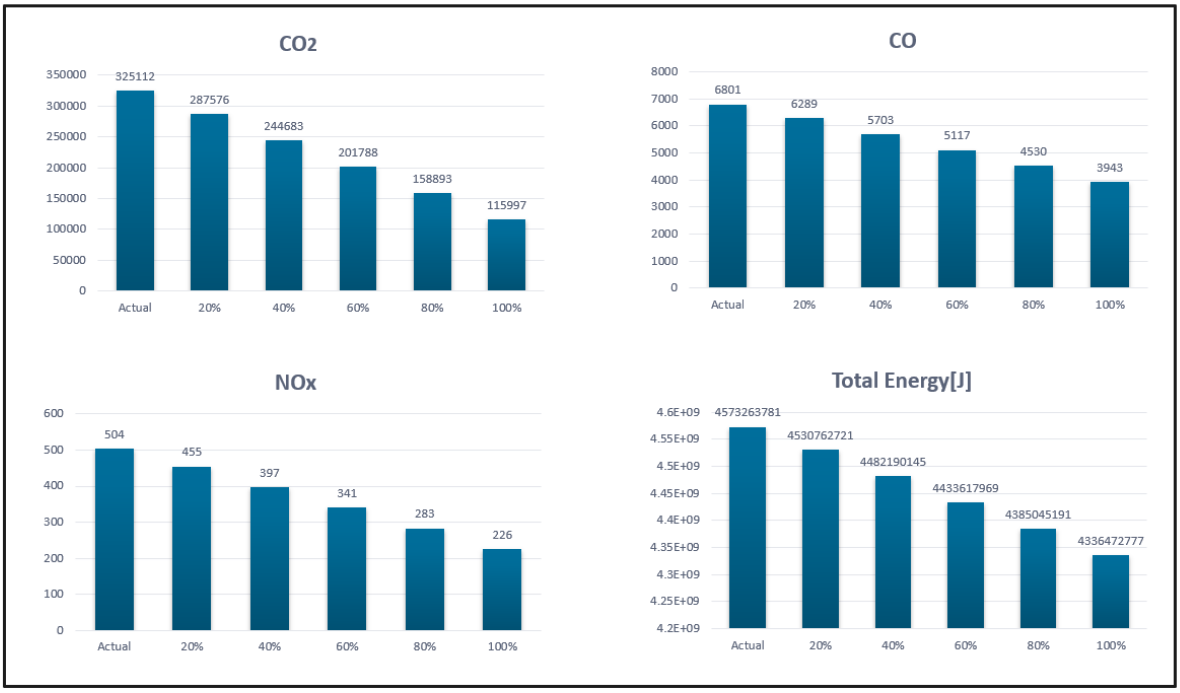


Figure 3.65 Pollution and Consumption VS Percentage of Electric Vehicle

From the result shown in **Fig. 3.65**, as increase of the percentage of electric vehicle, the pollution can be largely decreased by adopting of the electric vehicle, also the energy consumption is a little bit lower with electric car, the consumption for 100% electric vehicle is 6% lower than 2.5% of electric vehicle.

Source	Fuel	CO2	CO	Nox	TotalEnergy[J]	Distance[KM]
Motorcycle	Gasoline	89062	3836	98	1239272320	344.4
Passenger car	Electric	0	0	0	1286224384	244.6
Passenger truck	Electric	0	0	0	554149184	90.1
Light commercial truck	Electric	0	0	0	884123392	141.6
Transit bus	Gasoline	1090	5	0	15164326	1.6
Transit bus	Diesel	7671	20	32	104138624	6.4
Transit bus	Compressed Natural Gas	746	13	2	12641188	0.0
Refuse truck	Gasoline	159	7	1	2208637	0.0
Refuse truck	Diesel	16047	42	92	217853600	11.3
Refuse truck	Compressed Natural Gas	1222	20	1	20697122	1.6

Figure 3.66 Pollution and Consumption at 100% electric vehicle

MOVES treat the electric vehicle as a zero-emission transportation method, it is true only if all of the energy requested by electric vehicle are coming from the renewable energy source, considering the true energy structure of the Philippine, the benefit gain from the electric vehicle is overestimated, as mentioned in previous chapter, the usage of renewable energy is very important. The motorcycle will be the main source of pollution in table (**Fig. 3.66**), the shared micro-mobility service proposed

in **3.2 Overview of proposal** can replace the demand of motorcycle with the usage of electric bike or electric scooter, which can effectively alleviate the pollution by the motorcycle.
From energy point of view, the peak power generatable from the solar panel in our proposal will be 850 kw (**Fig. 3.13**), that means the 3,060,000,000J energy will be generated in one hour (3600 seconds), with respect to the electricity needed when there are 100% electric vehicle (2,724,496,960J from **Fig. 3.66**), all the demand electricity can be fulfilled with the solar panel, which means the benefit on pollution reduction from the electric vehicle can be maximized.

3.5 Traffic Scene and Virtual Environment Implementation

As mentioned in section **2.2.1 Definitions and Characteristics of Digital Twins**, the interoperability is an important characteristic and function of the digital twin, the digital world not only reflect the real world and showing what is happen, but also can interact with the user, in order to facilitate the information perception, test the solutions before its implementation, immerging in the digital world to training the operation process etc. a platform integrate with different data source (CAD digital model data, real-time data from IOT (internet of things), traffic data, infrastructure data, energy data etc.), physical engine to support the interaction system (response the user's input, interaction between scene objects etc.), the user interface etc. is necessary. In our case, the game engine Unity has been used to fulfill the requirements above mentioned.

3.5.1 Introduction to Unity

Unity is a cross-platform development tool developed by the British company Unity Technologies that integrates game development, real-time 3D animation creation, and architectural visualization. Unity is known for its cool 3D rendering effects and powerful cross-platform, it can easily develop gorgeous 2D and 3D content, and then publish it to multiple platforms with one click, Cross-platform saves the time and effort of developers required for platform migration, the differences between platforms have a great impact on the development progress of the project, for example, the differences in hardware, operation methods, screen sizes and other conditions will cause the problem on different platforms, so developers need to spend a lot of time and energy to do the porting and development between platforms. Unity has formed a complete ecological chain. Its asset store has various free and paid resources uploaded by developers and company in the community, developers can easily search in the asset store for everything need, such as delicate textures, realistic models, diverse plug-ins, cool animations, project cases and Unity tutorials, and these resources can be directly imported through Unity software, which facilitates the learning and use of developers. The characteristic of visual editing feature of Unity3D is "what you see is what you get", some related scenes are composed according to actual needs. Its advantages include a physics engine that gives users convenience to help realize physical phenomena such as gravity in the real world and realistic effects required in the scene, so it is more suitable for rapid development of simulation systems in virtual environments.

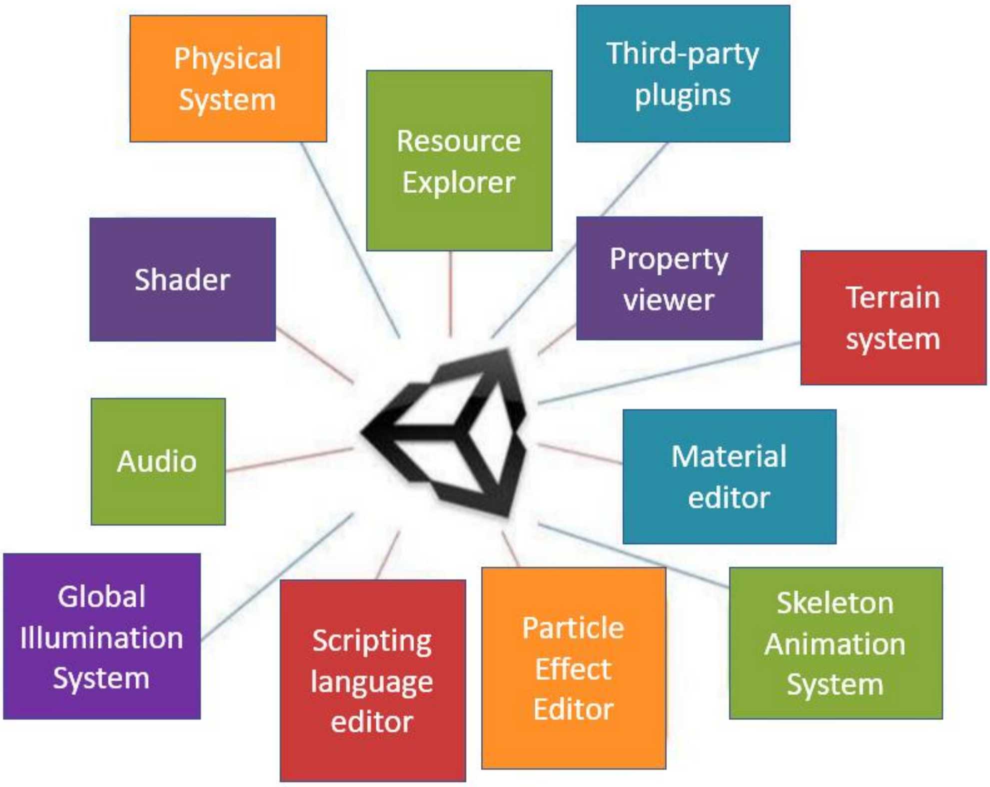


Figure 3.67 Unity Basic Modules

The basic modules of Unity are shown in **Fig. 3.67**, The physics system is a computer program that simulates the Newtonian mechanics model, it simulates and predicts the physical effects under various situation by controlling and changing conditions such as mass, friction, speed, and air resistance. Unity has a built-in PhysX physics engine developed by NVIDIA. PhysX can call the GPU to perform floating-point operations, so it can easily perform large-scale mathematical calculations, users can use PhysX to simulate various physical effects realistically and efficiently. [82]

The overall development framework of Unity3D is divided into four parts: application, component, game object, scene.

Applications are encapsulated applications with independent functions in Unity3D, such as rigid bodies, sky boxes, collision bodies, mountains, various lighting, etc. Developers can directly call these functions in the project, saving the time and effort to develop them separately. At the same time, according to different requirements, developers can also adjust the details of these functional modules to meet project requirements. Application is the lowest and indispensable module in the Unity3D development framework.

Component: each object in Unity is composed by various components and scripts, the component that every object must have is the Transform component (which determine the position and angles of the objects), and there are also optional components such as script, Mesh, Physics, Rendering Effects, Audio etc., developers can directly call related components to achieve some commonly used functions. However, only fine-tuning can be made when calling components. For some personalized requirements, it is difficult to realize them by simply call off-the-shelf components, at this time, scripts are used. Script is also a kind of component, but it is more flexible.

Game object is composed by components, it can be an object with various shapes and colors such as people, flowers, trees, buildings, water surfaces, cars, aircraft, etc., which containing many components and attributes, or it can be simply created as a empty objects with attached scripts which used to trigger certain functions. From this point of view, the scope of game objects is very wide. Any objects in the project can basically be called game objects, and various game objects eventually make up the project.

Scene is the virtual world created by the project. A virtual simulation project is composed of scenes, and the scene is composed by many game objects and various components attached to the game objects. [82]

The basic user interface of the Unity is showing in **Fig. 3.68**, which includes following modules.

Project: What is displayed in the Project view is the resource directory of this project, which is used to manage the resources used in the project (scene, 3d model, material, sound file, etc.). Right-click any file and select see in explorer, then the file folder location will be opened with the system File Explorer.

Hierarchy: The hierarchical view contains every virtual object in the current scene, including instances of resource files such as 3D models, as well as instances of prefabs. In short, the addition and deletion of instances will be reflected in the hierarchical view.

Scene: The scene view is the interactive sandbox system in the left. The scene view can be used to select and arrange the environment, characters, cameras, buildings and other scene objects.

Manipulating and moving game objects through the scene view is the most important function inside

Game: The game view is rendered from the camera in the virtual scene, and it represents the virtual world finally made by the developer. Among them, the display mode can be selected, and the data of rendering can also be displayed, which plays a certain role in the optimization of image performance.

Inspector: The inspector view displays the detailed information of the currently selected object, which can modify the relevant parameters of the object, and any properties displayed in the inspector view can be modified immediately. It can even modify the value of the script directly without modifying the script, and it can also modify some values while the project is running.

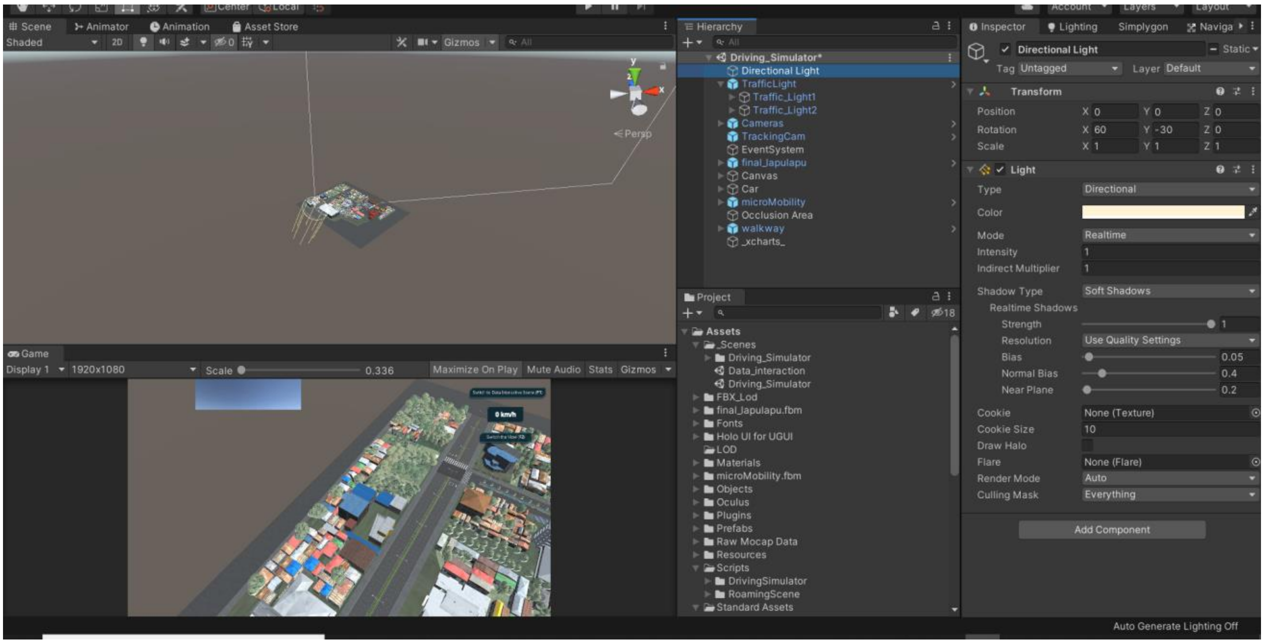


Figure 3.68 Basic User Interface of Unity3D

Unity uses C# and Unity Script (now obsolete) as the scripting language. C# language has cross-platform extensibility because it is built on .NET IL. In this way, game developers only need one set of code to run on multiple platforms (**Fig. 3.69**).

IL (Intermediate Language, CIL under the .NET platform, Common Intermediate Language) is an intermediate language format, similar to Java byte code. The code in this format requires a virtual machine to "interpret" and execute it. All IL instructions are based on the virtual stack: before calling the function, push the parameters to the virtual stack, when the function is executed, the parameters are taken out of the virtual stack, and then the result is pushed into the virtual stack. Due to the simple calling method, the instruction set of IL language is also relatively streamlined.

As a scripting language, IL can convert various features of the upper-level language of C# (such as generics, coroutines, etc.) into the basic IL instruction set, but such conversion also has a price, the converted IL instructions are more than ordinary function calls Out several times. [83]

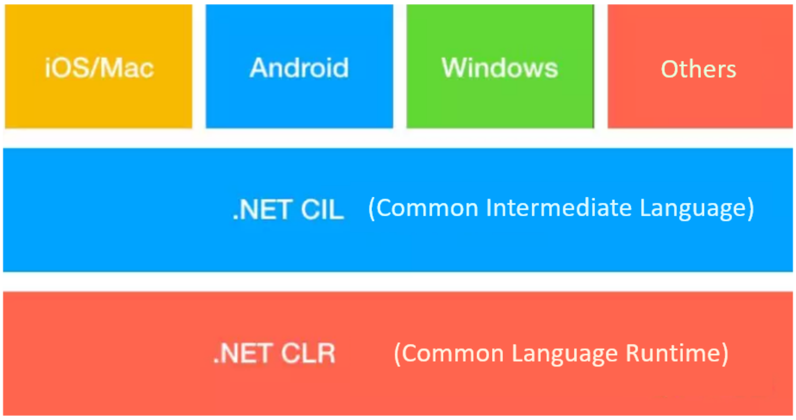


Figure 3.69 .Net CIL and CLR

In addition, IL language execution requires a virtual machine to translate into the machine code of the target platform. Although the .NET virtual machine is more efficient, there are still some gaps compared with the platform's native code, almost all game logic of Unity games is implemented through scripts. For a large project, thousands of scripts, the inefficiency caused by AOT packaging is a problem that has to be considered. Therefore, Unity introduced il2cpp technology in version 5.3.4. il2cpp is a tool for converting intermediate language into cpp code. The role of il2cpp is to remove the step of linking .NET CLR, "translate" the Managed Assembly generated by the C# script into C++ files, and finally compile these C++ files with the compiler of the target platform to generate the final game executable file. [83][84]

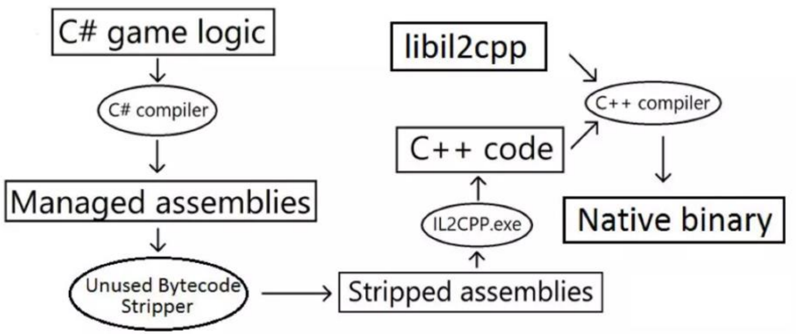


Figure 3.70 il2cpp working process

il2cpp will first read the .NET binary file, parse the symbols in it, and then convert the C# method into the corresponding C method. Although it is named il2cpp, it actually uses only a few C++ features, and most of the converted codes are C functions.

The Unity C# level interface is exposed to game developers. The developers write game logic through C# scripts, then translate the scripts into C++ files through il2cpp, and then link to the underlying C++ implementation of the Unity C# interface, and finally generate the binary files of the game. [84]

3.5.2 The road network information extraction from VISSIM

The traffic scene is built based on the co-simulation between Unity and VISSIM, where VISSIM generate the traffic related data which includes dynamic data such as the position and rotation of each vehicle in road network, the state of the traffic signal etc. and static data such as the position of each link, lane and connector, the position of the traffic signal etc. Unity should receive the vehicles related data and change each vehicle's status in the scene. Which request the road network in the scene of Unity is exactly same with the road network built in VISSIM, to guarantee the correctness of the position and rotation angle of each vehicle. In this section, the workflow of information extraction of road network in VISSIM will be described.

The road network data of VISSIM is stored in a file with format **.inpx**, if we double click this file, then the VISSIM will be opened and the microscopic traffic simulation can be run, but is we want to visualize the information saved in it, we should open it with the text editor.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<network version="702" vissimVersion="2020.00 - 12 [93405]">
    <anmDefaults colorDistr="1" crosswalkDisplayType="1" hgvPowerDistr="2" hgvWeightDistr="2" level="1" linkDisplayType
    ="1" panmFilename="" panmRoutesFilename="" pedestrianVehicleType="510" zoneConnectorDisplayType="1">
        <anmDefaultDrivingBehavior>
            <anmDefaultDrivingBehavior anmDrivBehavDefType="NONE" drivBehav="1"/>
            <anmDefaultDrivingBehavior anmDrivBehavDefType="URBAN" drivBehav="1"/>
            <anmDefaultDrivingBehavior anmDrivBehavDefType="RIGHTLEFTSIDE" drivBehav="2"/>
            <anmDefaultDrivingBehavior anmDrivBehavDefType="HIGHWAY" drivBehav="3"/>
            <anmDefaultDrivingBehavior anmDrivBehavDefType="PEDESTRIANS" drivBehav="4"/>
            <anmDefaultDrivingBehavior anmDrivBehavDefType="BICYCLELANE" drivBehav="5"/>
        </anmDefaultDrivingBehavior>
        <vehicleCategoryANMDefaults>
```

Figure 3.71 Information in .inpx File

As we can see from **Fig. 3.71**, the **.inpx** data are coded in XML. XML has several characteristics, one is plain text, which uses UTF-8 encoding by default, and the other is nestable, suitable for representing structured data. In addition, XML content is often transmitted as a message over the network.

XML has a fixed structure, the first line must be <?xml version="1.0"?>, and optional encoding can be added. Then, the document definition type (DTD: Document Type Definition) is declared, DTD is optional. Next is the content of the XML document. An XML document has one and only one root element. The root element can contain any number of child elements, and the element can contain attributes. For example, <network version="702" vissimVersion="2020.00 - 12 [93405]"> contains two attribute network version and vissimVersion, and the elements must be nested correctly.

```xml
    <links>
        <link assumSpeedOncom="60" consVehInDynPot="false" costPerKm="0" desSpeedFact="1" direction="ALL" displayType="
        1" emergStopDist="5" gradient="0" hasOvtLn="false" isPedArea="false" level="1" linkBehavType="6" linkEvalAct="
        true" linkEvalSegLen="10" lnChgDist="200" lnChgDistIsPerLn="false" lnChgEvalAct="true" lookAheadDistOvt="500"
        mesoFollowUpGap="0" mesoSpeed="50" mesoSpeedModel="VEHICLEBASED" name="" netPerfEvalAct="true" no="1" ovtOnlyPT
        ="false" ovtSpeedFact="1.3" showClsfValues="true" showLinkBar="true" showVeh="true" surch1="0" surch2="0"
        thickness="0" vehDynPotG="3" vehRecAct="true">
            <geometry>
                <linkPolyPts>
                    <linkPolyPoint x="-1.1861078508556204" y="-1.5814771413476016" zOffset="0"/>
                    <linkPolyPoint x="148.72601274277164" y="103.31678367894318" zOffset="0"/>
                </linkPolyPts>
            </geometry>
            <lanes>
                <lane width="3.5"/>
                <lane width="3.5"/>
                <lane width="3.5"/>
            </lanes>
        </link>
        <link assumSpeedOncom="60" consVehInDynPot="false" costPerKm="0" desSpeedFact="1" direction="ALL" displayType="
```

Figure 3.72 Example of a Link Element in XML

The **.inpx** includes any road networks information built in VISSIM, that means for each object we created during **3.3.4 Traffic simulation setup**, the relevant information about this object is saved in this file and coded in XML.

In C#, the XML content can be easily loaded by using the namespace *System.Xml*, this name space includes relative functions to operate the XML content. In our case, the interested information is the data relative to the position, size, shape of each links. Using the link in **Fig. 3.72** as example, the attributes configured in the simulation setup phase is included in the attributes of node link such as desired speed, driving behavior, display type etc. in the sub-element of node link, there are element geometry which specify the position coordinate of this link, and also the lanes elements which identify the number of lanes and width of each lane. If this link is formed by only one link segment, then there will be only 2 points (the start points and the end points) to identify this link, is the link has many link segments, then the number of elements “linkPolyPoint” will be more than two.

```xml
</link>
<link assumSpeedOncom="60" consVehInDynPot="false" costPerKm="0" desSpeedFact="1" direction="ALL" displayType="
1" emergStopDist="5" gradient="0" hasOvtLn="false" isPedArea="false" linkBehavType="6" linkEvalAct="true"
linkEvalSegLen="10" lnChgDist="200" lnChgDistIsPerLn="false" lnChgEvalAct="true" lookAheadDistOvt="500"
mesoFollowUpGap="0" mesoSpeed="50" mesoSpeedModel="VEHICLEBASED" name="" netPerfEvalAct="true" no="10000"
ovtOnlyPT="false" ovtSpeedFact="1.3" showClsfValues="true" showLinkBar="true" showVeh="true" surch1="0" surch2=
"0" thickness="0" vehDynPotG="3" vehRecAct="true">
    <fromLinkEndPt lane="6 1" pos="93.772920320097015"/>
    <geometry>
        <linkPolyPts>
            <linkPolyPoint x="219.42727319714308" y="38.593168028073265" zOffset="0"/>
            <linkPolyPoint x="232.30380714842019" y="49.057575447314662" zOffset="0"/>
        </linkPolyPts>
    </geometry>
    <lanes>
        <lane/>
    </lanes>
    <toLinkEndPt lane="7 1" pos="2.2439471587465034"/>
</link>
```

Figure 3.73 Example of a Connector Element in XML

As before mentioned, the connector in VISSIM is a special type of link, in the XML document, the connect and the link are same in most part of attributes and elements, the differences is the connector start from a link and finish to a link, so the number of the lanes and the width of each lane does not depends on the connector itself, but depends on the link where it come from and the link where it end to, the attribute “pos” is the relative position with respect to the start point position of the relevant links. That means the information of the connector is depending on the information of the link, we need read and store the information of all the link firstly in a loop, then the information of the connector can be gained completely in another loop by reading the XML content and the links data saved during last loop.

The relevant method in C# to read the XML content includes:

XmlDocument.GetElementsByTagName (Returns an XmlNodeList containing a list of all descendant elements that match the specified Name), *XmlDocument.LoadXml(String)* (load the XML document from the specified string), *XmlNode.Attributes[String]* (get the attribute of the name specified by string).

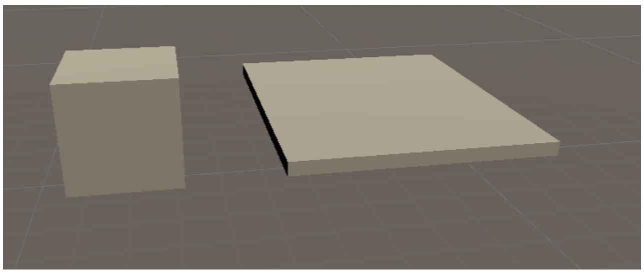


Figure 3.74 Cubes in Unity

In Unity, exist some basic shape such as cube, sphere, capsule, cylinder, plane etc. we can use these basic elements to form the shape more complex by combining them and change the factor of them in the inspector panel. The cube can be adjusted to a flatten board by changing the scale in x, y, and z direction (**Fig 3.74**), this flatten board can be seen as the lane of each link segment, and many of them combined together can form our road network. Where the scale in y direction is the height of the road surface, z direction scale is the length of the lane, and x direction scale is the width of the lane. The xyz coordinate is the lane center position. The direction of the lane should correspond to the direction of the link segment which the lane belongs to.

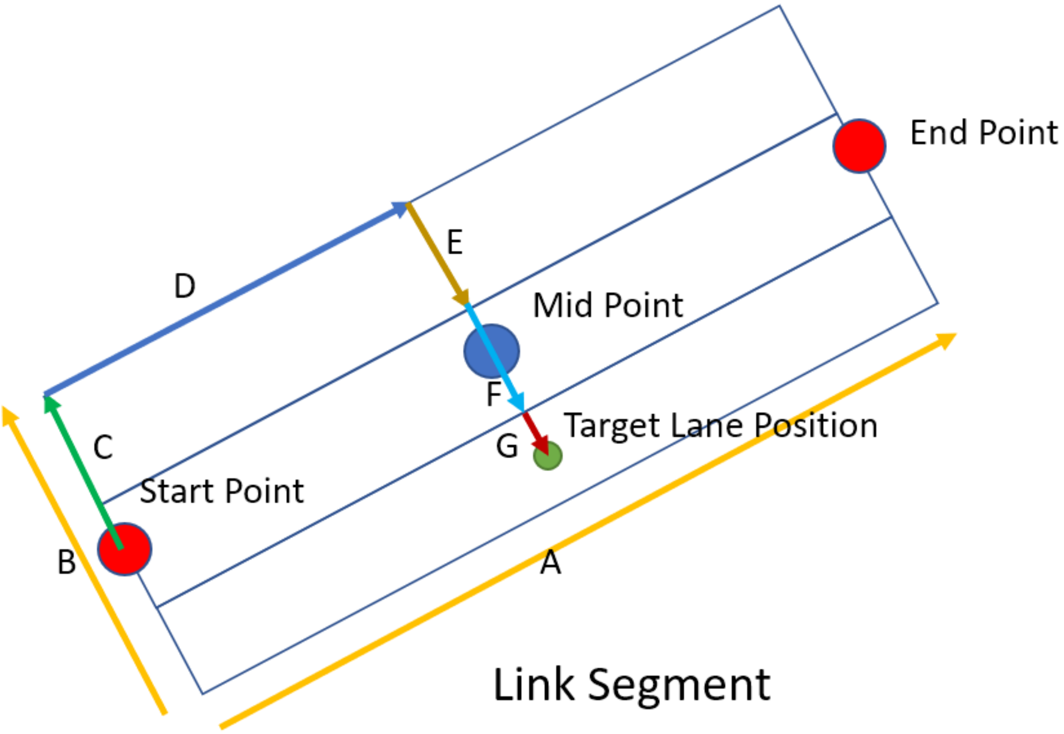


Figure 3.75 Method to Get Lane Center Position

Since the center position of each lane not indicated apparently in the XML document, we need to estimate it from known attributes. The attributes we can obtain from the file are the coordinate of poly points from the nodes “LinkPolyPoint” (**Fig. 3.72**), lane width from nodes “Lane” if the link segment belongs to a normal link (**Fig. 3.72**), if the link segment belongs to a connector, then lane width is get from calculating the average between “from link” and “to link”, the from and to links ID is coming from nodes “fromLinkEndPt” and “toLinkEndPts” (**Fig. 3.73**).

So, in **Fig. 3.75**, the known factors are the coordinate of start point, the coordinate of end point, and the width of each lane of the link segment. Our target is to calculate the coordinate of the “Target Lane Position”, firstly, the vector A can be calculated through:

$$\boldsymbol{A} = \boldsymbol{P_{End}} - \boldsymbol{P_{Start}} \ (4)$$

Then the vector B can obtain from (the coordinate system in Unity follows the Left-Hand rule where y towards up, z towards forward, and x towards right and vector B is perpendicular to vector A):

$$\boldsymbol{B} = (-\boldsymbol{Z}_A , \boldsymbol{0} , \boldsymbol{X}_A) \quad (5)$$

Then B normalized equal to (suppose the lane from top to bottom are 1, 2, 3 respectively and W is width of the lane):

$$\boldsymbol{B}_{norm} = \frac{\boldsymbol{B}}{\boldsymbol{W}_1 + \boldsymbol{W}_2 + \boldsymbol{W}_3} \quad (6)$$

Vector C has the same direction with vector B, its length is equal to the half of the total width of the link segment:

$$\boldsymbol{C} = \frac{\boldsymbol{B}}{2} \quad (7)$$

Vector D has the same direction with the A, its length is half of A:

$$\boldsymbol{D} = \frac{\boldsymbol{A}}{2} \quad (8)$$

Vector E, F, G have the direction opposite to B with width of lanes or half width of lanes:

$$\boldsymbol{E} = -\boldsymbol{B}_{norm} \times \boldsymbol{W}_1 \ ; \ \boldsymbol{F} = -\boldsymbol{B}_{norm} \times \boldsymbol{W}_2 \ ; \ \boldsymbol{G} = -\boldsymbol{B}_{norm} \times \frac{\boldsymbol{W}_3}{2} \quad (9)$$

Finally, the coordinate of the target point (center of the target lane) is:

$$\boldsymbol{P}_{Target} = \boldsymbol{P}_{Start} + \boldsymbol{C} + \boldsymbol{D} + \boldsymbol{E} + \boldsymbol{F} + \boldsymbol{G} \quad (10)$$

Using this method, we can have the width, length, orientation, and center position of each lane.

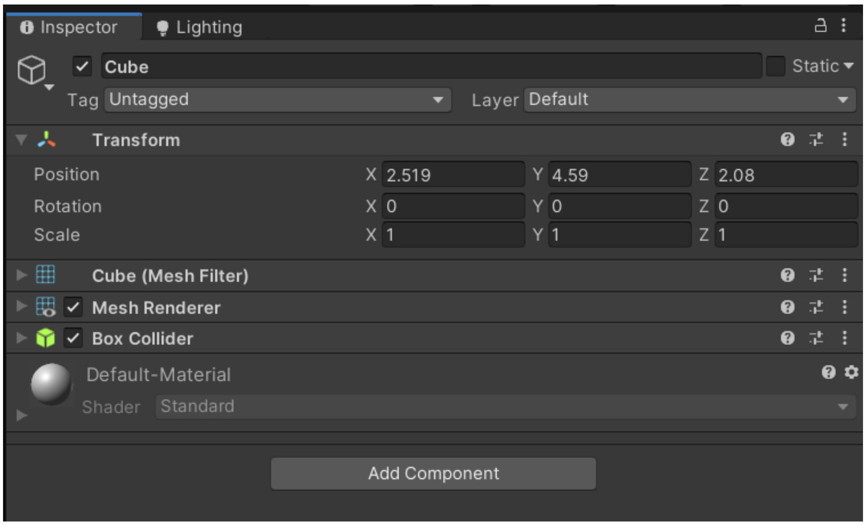


Figure 3.76 Transform Component of Object in Unity

These properties can be set in the transform component of the cube, the position is the xyz coordinate of the lane center, the orientation is configured with the rotation (rotation in inspector panel is in unit grade), the length and width can be set with the scale, the scale unit is meter, as above mentioned, y is the upward direction in unity, so width and length are corresponding to x and z. Of course, this mission should accomplish by the C# script because there are hundreds of lanes need to be generated.

To keep the codes explanation short and easy to understand, the pseudo codes will be used in this thesis, the script of the road network generation in Unity is shown below:

Load the .inpx file
doc = load(VISSIMFileName.inpx)
Loop to process the link element in XML file one by one
for link in ListOfAllLinkElements:
    Check if this link is a connector
    if (link contains sub-element "fromLinkEndPt" or "toLinkEndPt"):
        Pass current loop cycle
        continue
    Read the Lanes in this link and save them properties
    laneIndex = 1
    for lane in ListOfLanesInThisLink:
        lane.id = laneIndex
        lane.width = lane.attributes["width"].value
        laneIndex+1
        add this lane in lanes
    link.id = link.attributes["no"].value
    Put all coordinates in a list
    for coordinate in AllCoordinateIn("linkPolyPts")
        coordinates.append(coordinate)
    Read the list and save the coordinate to link Segment
    linkSegmentIndex = 1
    for coordinate in coordinates
        linkSegment.id = linkSegmentIndex
        linkSegment.startPoint = (linkSegmentIndex)th coordinate in coordinates
        linkSegment.endPoint = (linkSegmentIndex+1)th coordinate in coordinates
        add this linkSegment in linkSegments
        linkSegmentIndex + 1
    Save the link segments and lanes in this link
    link.lanes = lanes

link.linkSegments = linkSegments

add this link into links

Now we have the links dictionary which contains all no-connector links information, similarly, with another loop for the connector, by using the link id, we can find the "from" and "to" link of this connector in the links dictionary, then through the lanes dictionary saved in "from" and "to" link and the "from" and "to" index of the lanes for this connector, we can find the widths of the "from" and "to" lanes, by averaging the "from" and "to" lane, the width of the lanes in connector can be calculated, and then we should add the connectors into the links dictionary too.

Loop to process the link element in XML file one by one

for link in ListOfAllLinkElements:

Check if this link is a connector, if not, then do nothing in current loop cycle

if (link contains sub-element "fromLinkEndPt" or "toLinkEndPt"):

It is a connector

connector.id = link.attributes["no"].value

Read the "from" and "to" link and lanes of this connector

connector.fromLinkNo = link.nodes("fromLinkEndPt")[0].attribute["lane"].value.split(" ")[0]

connector.fromLaneNo = link.nodes("fromLinkEndPt")[0].attribute["lane"].value.split(" ")[1]

connector.toLinkNo = link.nodes("toLinkEndPt")[0].attribute["lane"].value.split(" ")[0]

connector.toLaneNo = link.nodes("toLinkEndPt")[0].attribute["lane"].value.split(" ")[1]

connector.laneCount = link.nodes("lane").count

Put all coordinates in a list

For coordinate in AllCoordinateIn("linkPolyPts")

coordinates.append(coordinate)

Read the list and save the coordinate to link Segment

linkSegmentIndex = 1

for coordinate in coordinates

linkSegment.id = linkSegmentIndex

linkSegment.startPoint = (linkSegmentIndex)th coordinate in coordinates

linkSegment.endPoint = (linkSegmentIndex+1)th coordinate in coordinates

add this linkSegment in linkSegments

linkSegmentIndex + 1

Save the link segments in this connector

connector.linkSegments = linkSegments

add this connector into links

Iterate all link in links with type connector

```
for connector in links:
    fromWidth = links[connector.fromLinkNo].lanes[connector.fromLaneNo].width
    toWidth = links[connector.toLinkNo].lanes[connector.toLaneNo].width
    Calculate connector lane widths from the fromWidth and toWidth
    for(i = 1;i <= connector.laneCount; i++):
        lane.width = (fromWidth+toWidth) / 2
        lane.ID = i
        add lane to connector.lanes
```

Now information needed to build the road network in Unity is in dictionary links, we can start to draw the lanes

Iterate all links

```
for link in links:
    Iterate all link segment in this link
    for linkSegment in link.linkSegments:
        vectorA = linkSegment.startPoint – linkSegment.endPoint
        length = (linkSegment.startPoint – linkSegment.endPoint).magnitude
        Iterate and generate each lane in this link segment
        for lane in link.lanes:
            laneCenter =  (Calculate coordinate of lane center through equations (4) – (10))
            width = lane.width
            Create cube game object in Unity and set its transform
            instantiate("cube")
            cube.transform.position = laneCenter
            cube.transform.rotation = quaternian.lookRotation(vectorA)
            cube.scaleX = width
            Suppose the road surface has height 10 cm
            cube.scaleY = 0.1
            cube.scaleZ = length
```

From the script, by running the game in Unity, the road network is successfully generated in Unity, the position and the orientation are perfectly fit to the original road network drawn in VISSIM, the only difference is the width of the connector, in VISSIM, the width of connector is gradually changed from the "from" lane of the "from" link to the "to" lane of the "to link", while in Unity, since the lanes are generated from cubes, the width is the same all over the connector with the average value between the widths of “from” and “to” lanes.

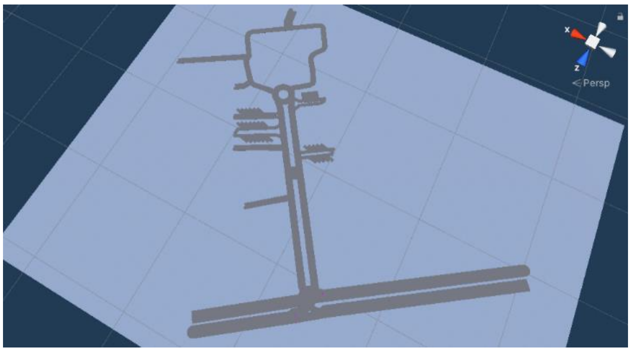


Figure 3.77 Road Network Generated in Unity

The result is shown in **Fig. 3.77**, the road network information from the VISSIM has been extracted and converted in 3D model in the Unity. In the next section, take the road network model built in this section as reference, the digital urban area model will be jointly made by 3ds MAX and CityEngine.

3.5.3 Urban area modelling

In last section, a simple road network model was built with Unity, it formed by extending and squash of many cubes, to make the traffic scene more real, the road surface, the road mark, and the surround building need to be added. Since Unity is not a tool for modeling, this process needs done with others software, the workflow of urban area implementation shown as **Fig. 3.78**.

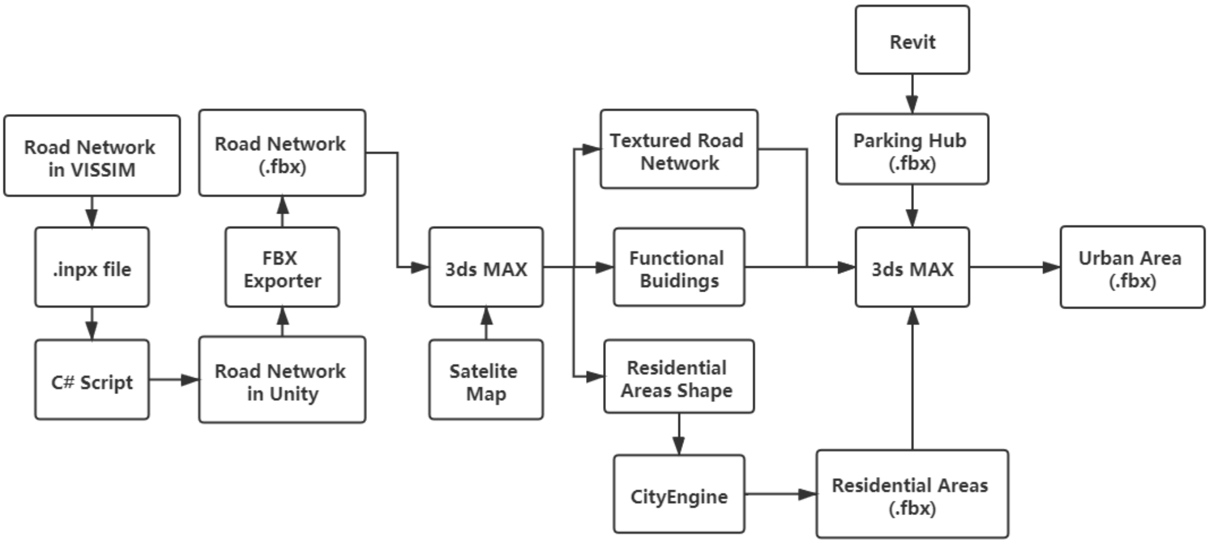


Figure 3.78 Workflow of Urban Area modelling

The road network model in Unity exportation is done with the plugin “FBX exporter”, FBX model is a universal model format that supports all major 3D data elements and 2D, audio and video media elements. Autodesk FBX is a cross-platform free 3D creation and exchange format software produced by Autodesk. FBX users can access 3D files from most 3D vendors.

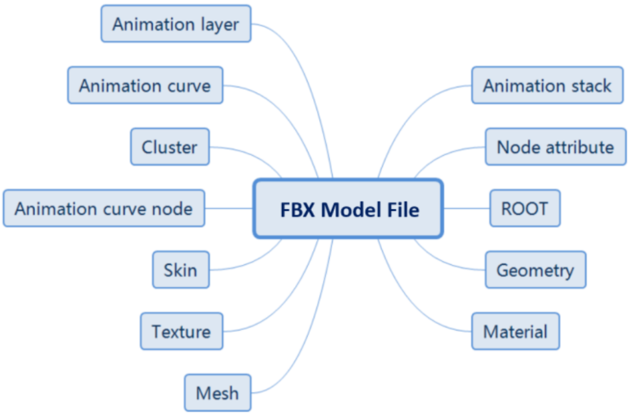


Figure 3.79 Content in FBX Model File

In our case, the .fbx model format has been used for different modeling software, even if each software has its own model format, the .fbx is well supported by all of them.

3D Studio Max, often referred to as 3DS Max or MAX, is a PC-based 3D animation rendering and production software developed by Discreet (later merged by Autodesk). 3DS Max has applications in film, interior design, engineering manufacturing, industrial graphics, virtual reality, and other fields.

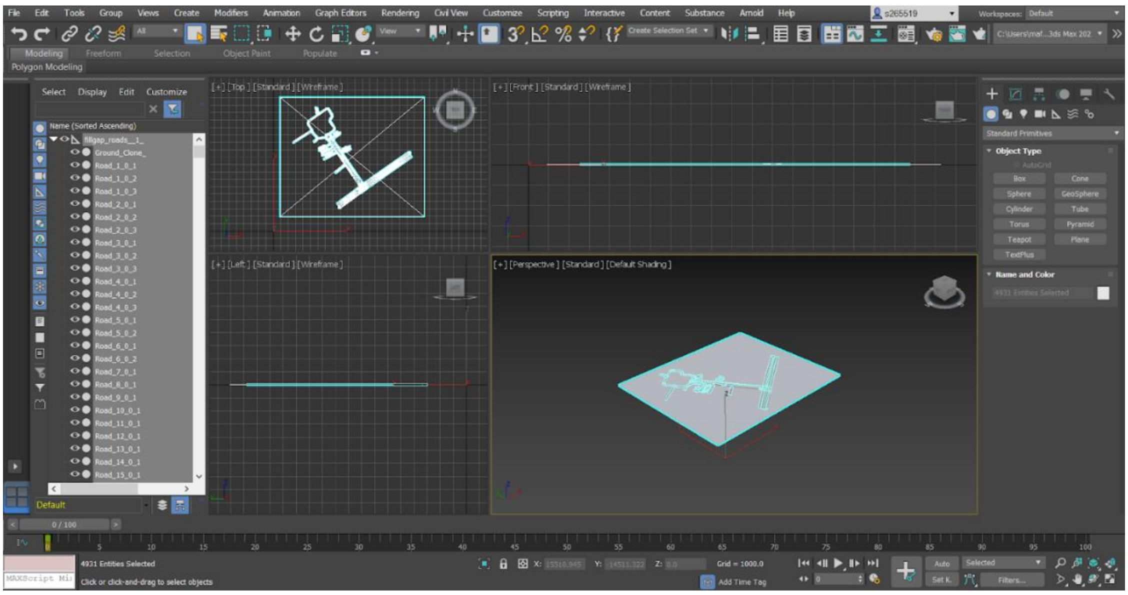


Figure 3.80 Road Network in 3ds MAX

After the road network exported from unity, we should import the .fbx file into 3ds MAX by selecting “import” in the file menu at the top left part of user interface of 3ds MAX (**Fig. 3.80**).

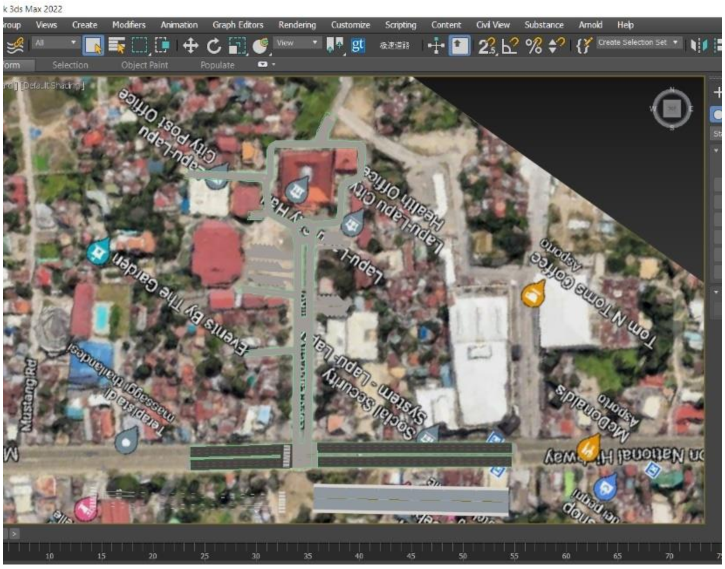


Figure 3.81 Road Network and Satellite Map as Reference Background

Firstly, the correctness of the unit of the road network should be checked, then we can use this road network as a reference to generate the urban area model. Of course, only with the road network is not enough to correctly place the building and different area, we need import a satellite map as a background for place reference of buildings, since in 3ds MAX does not exist the online map, we do not have the automatically calibrated map background like **Fig. 3.21** in VISSIM, one way to make the map background is to make a screenshot of satellite map in web browser, create a plane in 3ds MAX, then using the screenshot picture as the texture of this plan, to avoid the distortion of the map, we can set the length and width of the plane equal to the pixel length and width of the picture. Since the road network is drawn in VISSIM by taking the online map as the reference, its size should be fit to the reality, so we can adjust the plane size and orientation to fit the road network model (the length width ratio of the plane should maintain fixed during adjustment), the effect after adjustment is shown in **Fig. 3.81**, by selecting and right clicking the background objects, we can select the “freeze selection” to make these objects un-selectable and unmovable which avoid the mistake during the modelling process.

The road network imported is formed by hundreds of lanes like shown in scene explore panel (left panel of **Fig. 3.80**), which is not suitable from performance point of view, also it is hard to apply the textures because each road segment is not formed by a single surface, so the roads have been remodeled with some simple planes take the original road network as reference.

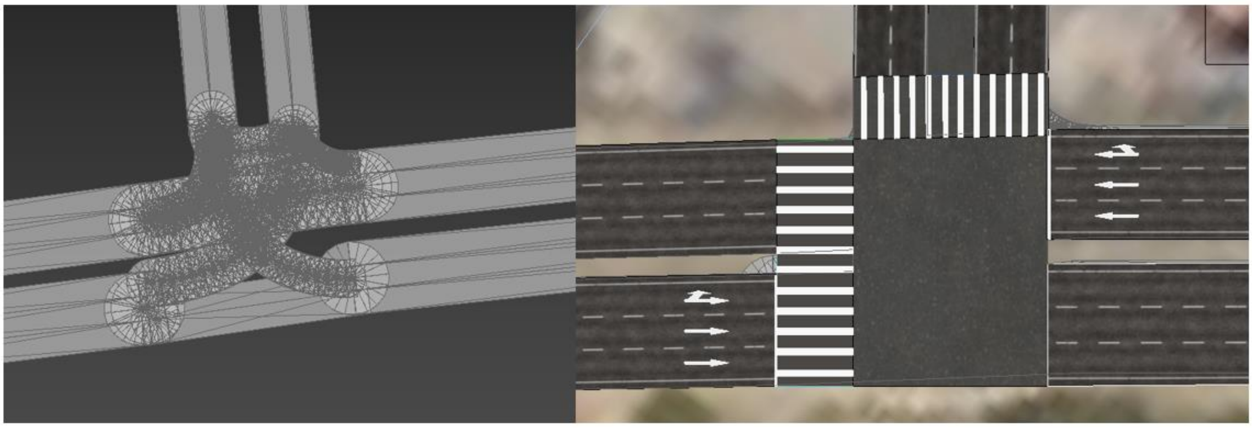


Figure 3.82 Before and After Remodeling of Roads

In **Fig. 3.82** we can see the remodeled roads become simpler and easier to apply the texture, the texture can be adjusted in slate material editor panel by press short cut button “M”.

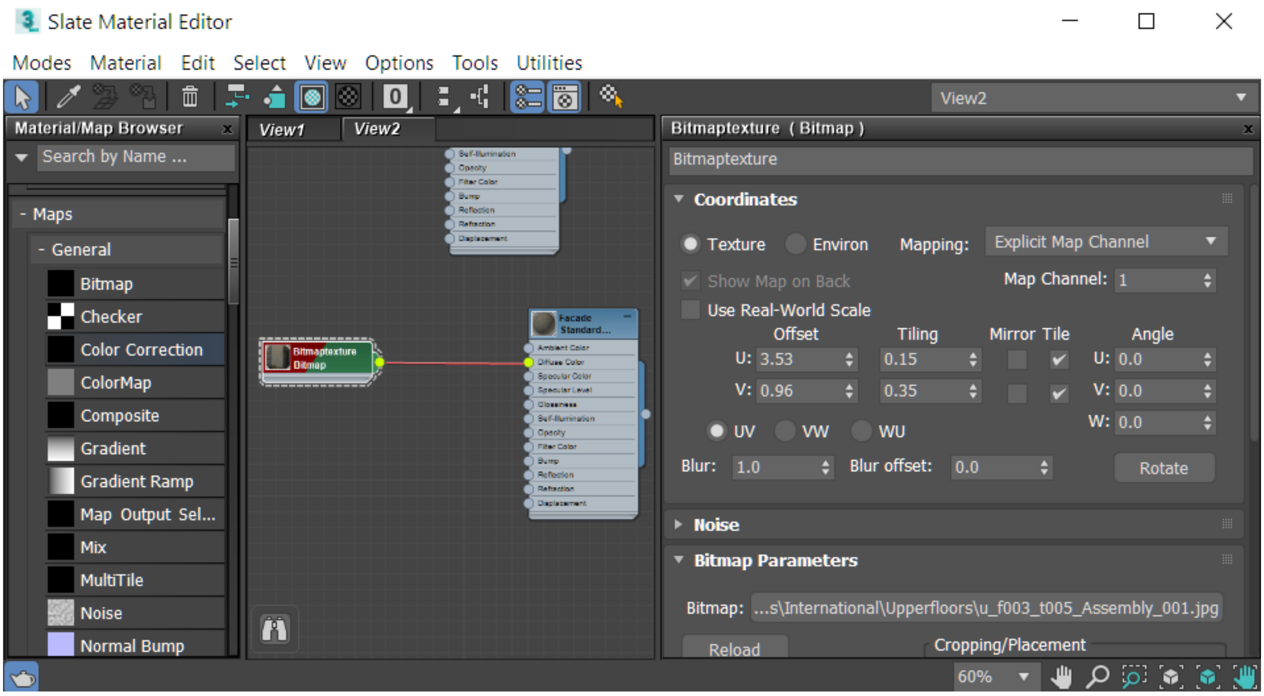


Figure 3.83 Slate Material Editor

Left part of **Fig. 3.83** shows the material/map browser. We can drag the material or texture from the Material/Map Browser panel to the view to edit them. The view panel allow us to edit and visualize the relationship between materials and textures. When we select a texture in view panel, on the right panel we can edit the offset, scale, repeat or not, angle etc. of the texture.

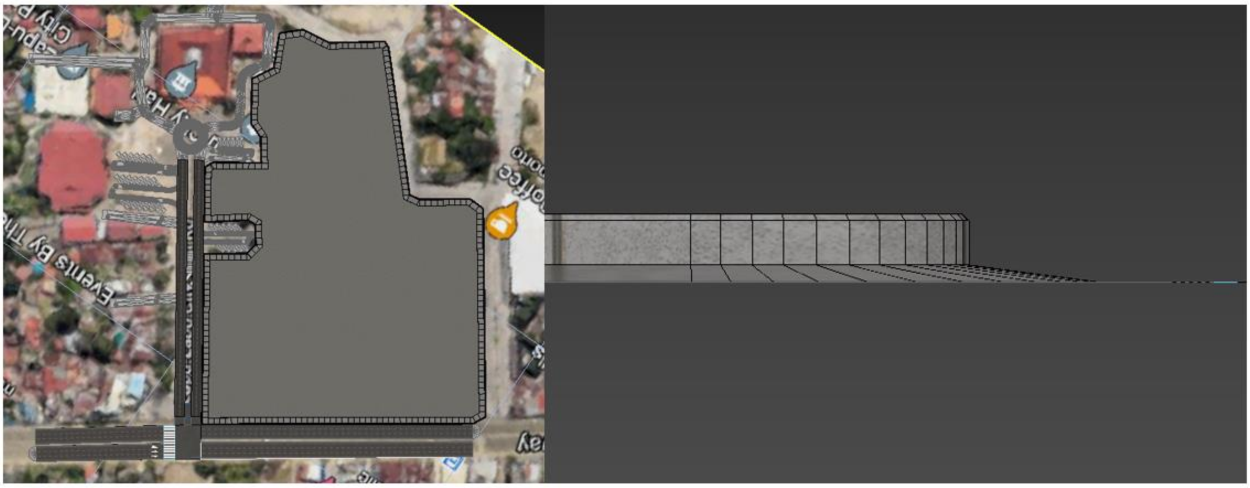


Figure 3.84 Ground and Curb

Next step is drawing the ground area, select the “line” in “shape” panel and then we can draw the line freely in top view, to extrude the shape drawn, the line must close to an area, then right click it, select convert to poly, in the modify panel, we can edit the faces include in poly and extrude it to a curb. For the shape of the curb (right of **Fig. 3.84**), we can draw the outline of ground with the “line”, then draw an area with the shape of the curb and convert it to editable poly, then select the extrude to spline and pick the outline as the path, then the curb with any shape can be done.

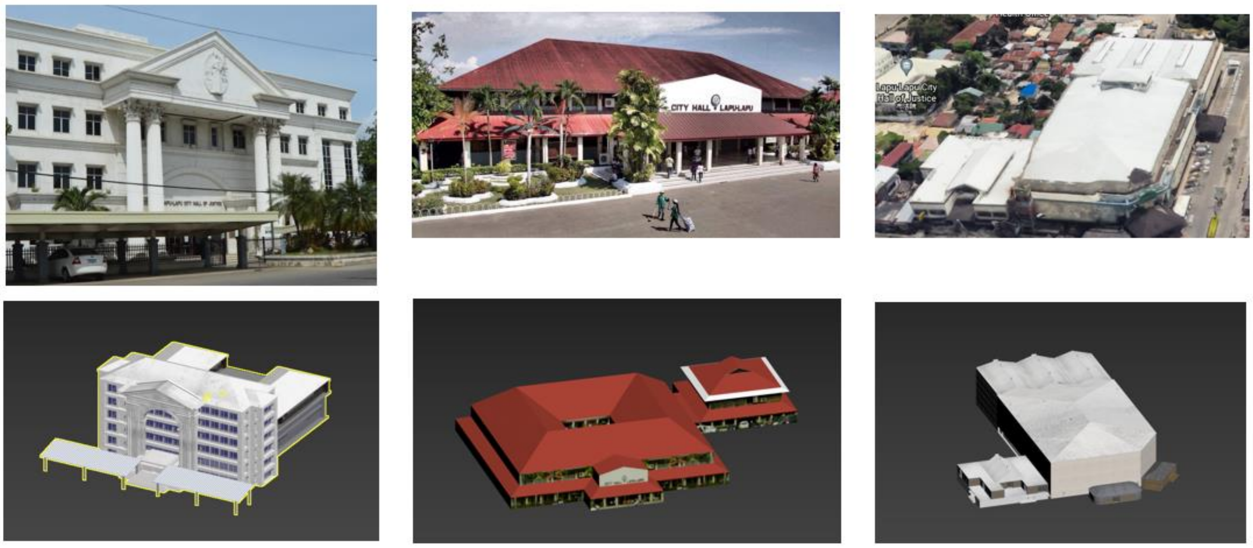


Figure 3.85 Functional Buildings

For the buildings important (usually has some special function or with big volume such as the offices, services point, hotel, shop etc.), the building shape information are obtained from the google earth (3D satellite map) or the photos. **Fig. 3.85** shows some examples of functional buildings, they are City Hall of Justice, Lapu-Lapu city hall, Gaisano mall respectively from left to right.

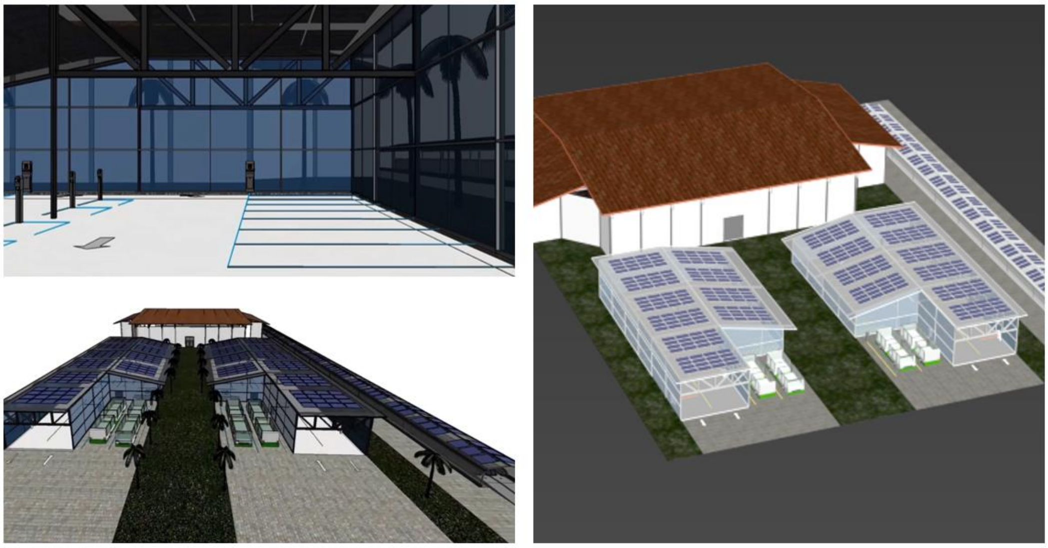


Figure 3.86 Parking Hub in Revit (Left) and imported in 3ds MAX (Right) (From Daniele)

The parking area and the walkway are designed by Civil Engineer student Daniele with software Revit, this area is modeled in very detail way because it is the area of smart mobility hub which we proposed before. As we can see from **Fig. 3.86**, the model exterior has changed after exporting it to .fbx format from Revit and then imported into 3ds MAX, but the model structure maintained the same.

Figure 3.87 The Manually Modeled Buildings

For the established model in 3ds MAX, the methods used are the similar. Use lines to outline the basic floor plane, convert the entire line into editable polygons, and extrude the corresponding faces, so that a simple floor can be made. The establishment of a house model is slowly produced from simple to complex. The first is to generate points, then points to generate lines, lines to generate surfaces, and then use surfaces to generate 3D models, this is the basic process of a building 3D model establishment. But the texture needs to prepare from the building characteristic of the area.

Figure 3.88 The Residential Area (Google Earth)

For the residential area with many houses (**Fig. 3.88**), modelling them manually one by one is very time consuming. By observing the area, all these residential houses have similar shape and characteristic, so we used another software CityEngine, which can generate many houses with script.

CityEngine was designed and developed in 2001 by Pascal Miller, PhD of Computer Vision Laboratory in ETH Zurich, and the company Procedural was established in 2007. In July 2008, the first commercial version of CityEngine2008 was released. In July 2011, Esri announced the acquisition of Procedural. Current CityEngine application areas include geographic information, urban planning, rail transit, cultural heritage, simulation, game development, movie entertainment etc.

The biggest advantage of CityEngine is that it can save a lot of time and cost for the construction of a large number of large-scale models of urban 3D scenes using batch modeling based on rules. At the beginning of modeling, it takes a certain amount of time to write rule files. When rule files reach a certain scale, for a large number of models and design schemes, CityEngine modeling speed will be much faster than traditional manual modeling. The established rules are directly dragged and dropped into the scene to be modeled, and a city-level 3D scene model can be established instantly. Once the model rules are established, they can be reused. [86] In our case, only few functions of CityEngine have been used.

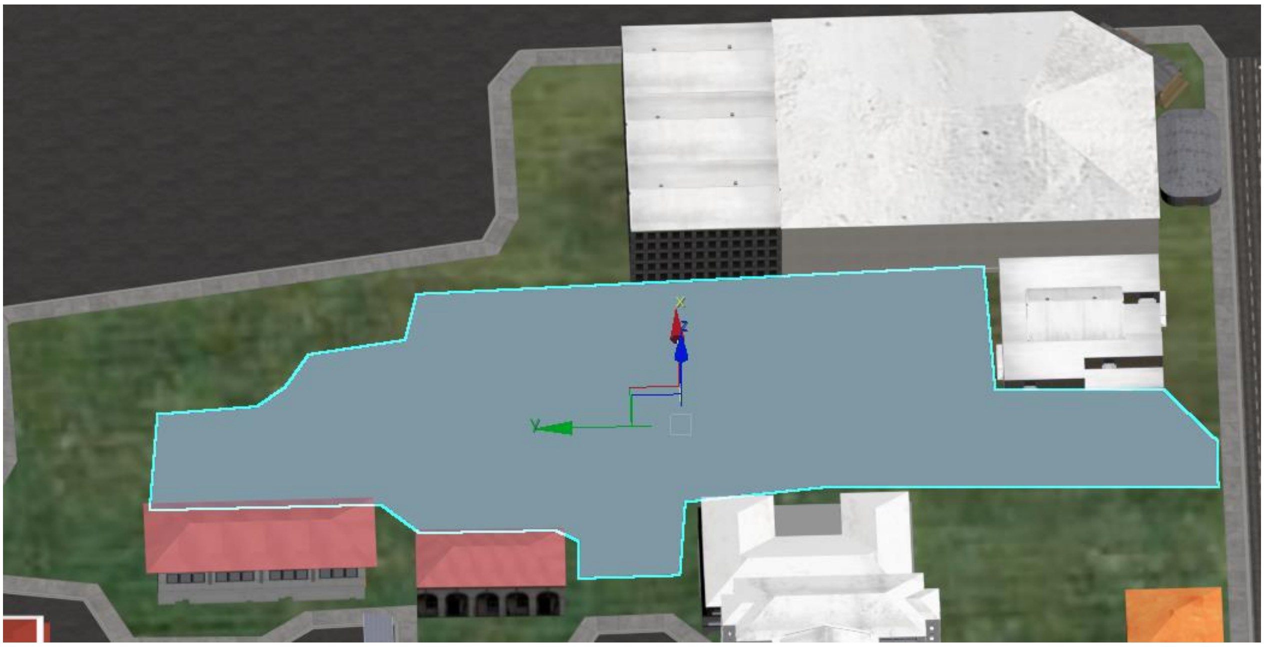


Figure 3.89 The Residential Area in 3ds MAX

Firstly, the residential area has been drawn with a close line take the map plane in **Fig. 3.81** as reference and then convert it into a polygon, after that, export this plane in to a FBX model by selecting export selected in file menu.

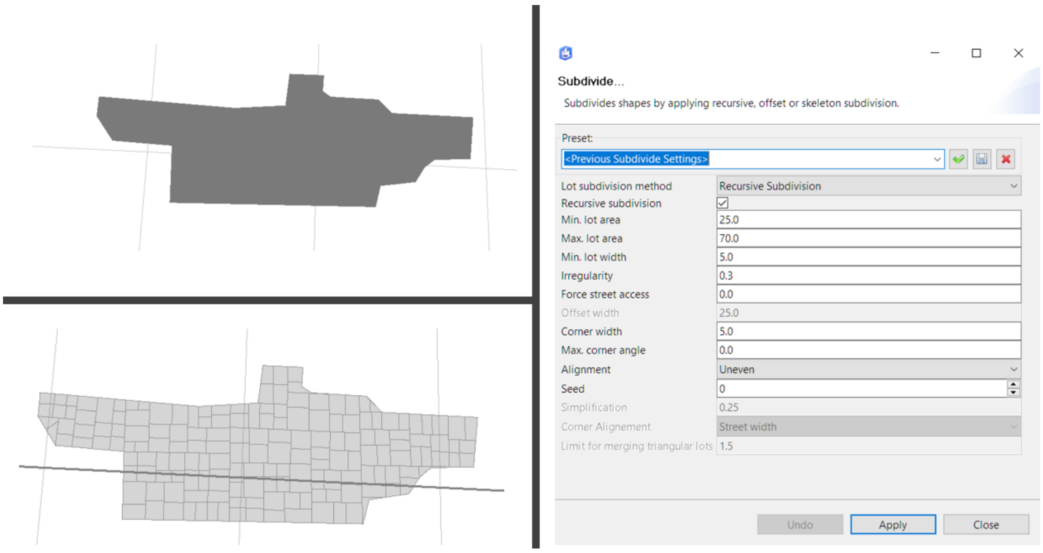


Figure 3.90 Subdivide area in CityEngine

In CityEngine, the FBX model been imported and draw the shape take the plane as the reference, in CityEngine, point and line does not exist, the basic geometry is the plane, by selecting “Polygonal Shape Creation” in the toolbar, any shape of the plane can be drawn. Next step is to select the plane, and subdivide it in “Subdivide Panel”, here we can configure the area range for each house block, then software will subdivide the area in sub-areas randomly.

The modeling in CityEngine is based on CGA rule, The full name of CGA is Computer Generated Architecture, which contains a series of rules and grammar to determine how the model is generated. The language used in the rules is called CGA Shape Grammar language. The script in CGA which help us to generate the houses of residential buildings of the area and the explanation of codes will be shown below (the source code will be shown since the CGA script is already short and easy to understand).

Import the texturing script from ESRI library (the roof and side textures should be prepared in corresponding library folder)

```
import Roof_Textures:"/ESRI.lib/rules/Roofs/Roof_Textures.cga"
import Facade_Textures:"/ESRI.lib/rules/Facades/Facade_Textures.cga"
```

Starting Script

```
@StartRule
```

Distance between the buildings equal to 2*setBackDis

```
attr setBackDis = 1
```

Building height set randomly between 3m and 9m (from observation of the area)

```
attr h = rand(3, 9)
```

Specify the textures for green area

```
attr gardenTex = fileRandom("Texture/Garden/"+"*.jpg")
```

The height of the roof set randomly between 1-3 m

```
attr Roof_Ht = rand(1,3)
```

Divide the sub-areas to building use or green area (ratio can be changed depending on the situation)

```
Lot-->
    80%: building
    else: garden
```

The sub-areas been marked “building” need to setback for a distance, the gap between buildings be marked as “garden”

```
building-->
    setback(setBackDis) { front : garden  | remainder : buildShape}
```

Pass the "buildShape" areas to a function which splits concave polygons into sub-polygons

```
buildShape-->
    Extrusion(h,1)
Extrusion(height,maxLength) -->
    convexify(maxLength)
    comp(f){ all: alignScopeToGeometry(yUp, 0, longest) ExtrusionConvexified(height,maxLength) }
```

Extrude areas to form blocks

```
ExtrusionConvexified(height,maxLength) -->
    case scope.sx < maxLength+1 || scope.sz < maxLength+1: NIL
    else:
        extrude(height)
        floorBuilding
```

Divide the block into top face and side face

```
floorBuilding-->
    comp(f){side:sidef|top:topf}
```

Top face will be assigned randomly to one roof type based on the probability specified

```
topf-->
    20%: ShedRoof
    20%: PyramidRoof
    15%: GableRoof
    15%: HipRoof
    5%: HalfHipRoof
    5%: GabletRoof
    5%: GambrelRoof
    5%: MansardRoof
    5%: GambrelFlatRoof
    else: MansardFlatRoof
```

Generate corresponding roof

```
ShedRoof -->
    roofShed(15) RoofMassScale
GableRoof -->
    roofGable(45,0,0,false,0) RoofMassScale
HipRoof -->
    roofHip(45) RoofMassScale
PyramidRoof -->
    roofPyramid(45) RoofMassScale
HalfHipRoof -->
```

```
    roofGable(45,0,0,false,0) s('1,Roof_Ht,'1)
    split(y){ '0.5: RoofMass(true) comp(f){ bottom: NIL | horizontal: set(Roof_Ht,Roof_Ht*0.5)
HipRoof } }
GabletRoof -->
    roofHip(45) s('1,Roof_Ht,'1)
    split(y){ '0.5: RoofMass(true) comp(f){ bottom: NIL | horizontal: set(Roof_Ht,Roof_Ht*0.5)
GableRoof } }
GambrelRoof -->
    roofGable(70,0,0,false,0)
    split(y){ Roof_Ht*0.7: RoofMass(true) comp(f){ bottom: NIL | horizontal:
set(Roof_Ht,Roof_Ht*0.3) Gab } }
MansardRoof -->
    roofHip(70)
    split(y){ Roof_Ht*0.7: RoofMass(true) comp(f){ bottom: NIL | horizontal:
set(Roof_Ht,Roof_Ht*0.3) Hip } }
GambrelFlatRoof -->
    roofGable(45,0,0,false,0)
    split(y){ Roof_Ht: RoofMass(false) }
MansardFlatRoof -->
    roofHip(45)
    split(y){ Roof_Ht: RoofMass(false) }
RoofMassScale -->
    s('1,Roof_Ht,'1)
    RoofMass(false)
```

Remove both bottom and top face or just bottom

```
RoofMass(removeBottomAndTop) -->
    case removeBottomAndTop:
        comp(f){ horizontal: NIL | vertical: sidef | all: roofPlane }
    else:
        comp(f){ bottom: NIL | vertical: sidef | all: roofPlane }
upperBuilding-->
    comp(f){side:sideu|top:roofPlane}
```

Assign the building textures to the upper side face of the house

```
sideu-->
    Facade_Textures.Generate
```

Assign the roof textures

```
roofPlane-->
```

Roof_Textures.Generate

Assign the garden textures, make a copy of the area

```
garden-->
	setupProjection(0, scope.xz, ~20, ~20)
	texture(gardenTex)
	projectUV(0)
	copy.
	copy1
```

Transform the copy into scatter points which distribute in the area based on uniform distribution

```
copy1 -->
	scatter(surface,4,uniform) { trees }
```

Transform the scatter points into tree model

```
trees -->
	i("/city_hall/models/low poly tree.obj")
	s(8,10,8)
	center(xz)
```

Assign the textures to ground floor of the houses

```
sidef-->
	Facade_Textures.Generate
```

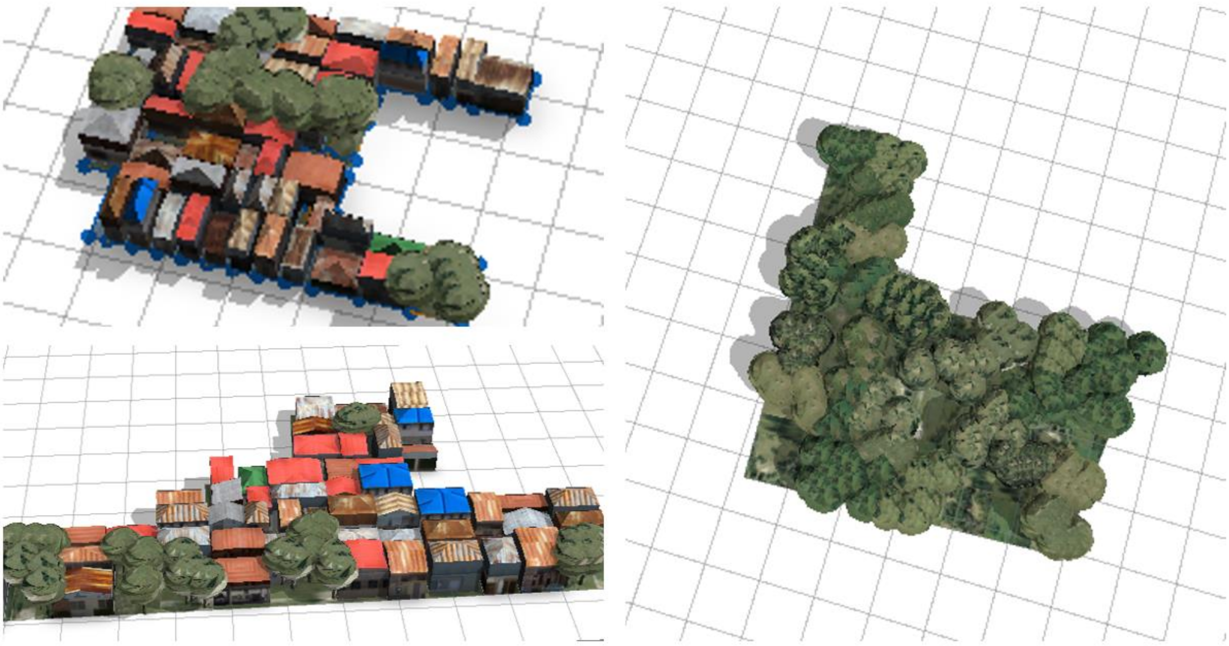


Figure 3.91 Houses Generated with Script

After the script has been wrote, the CGA rule file has be assigned to the subdivided area shape, by clicking “Generate” in toolbar, the different type of residential area can be generated (**Fig. 3.91**) by changing the variables in script, for example, if we set the area 0% building and else garden, then we can obtain a forest without houses.

Now we need export the houses batch from CityEngine in format .fbx and then import it into 3ds MAX

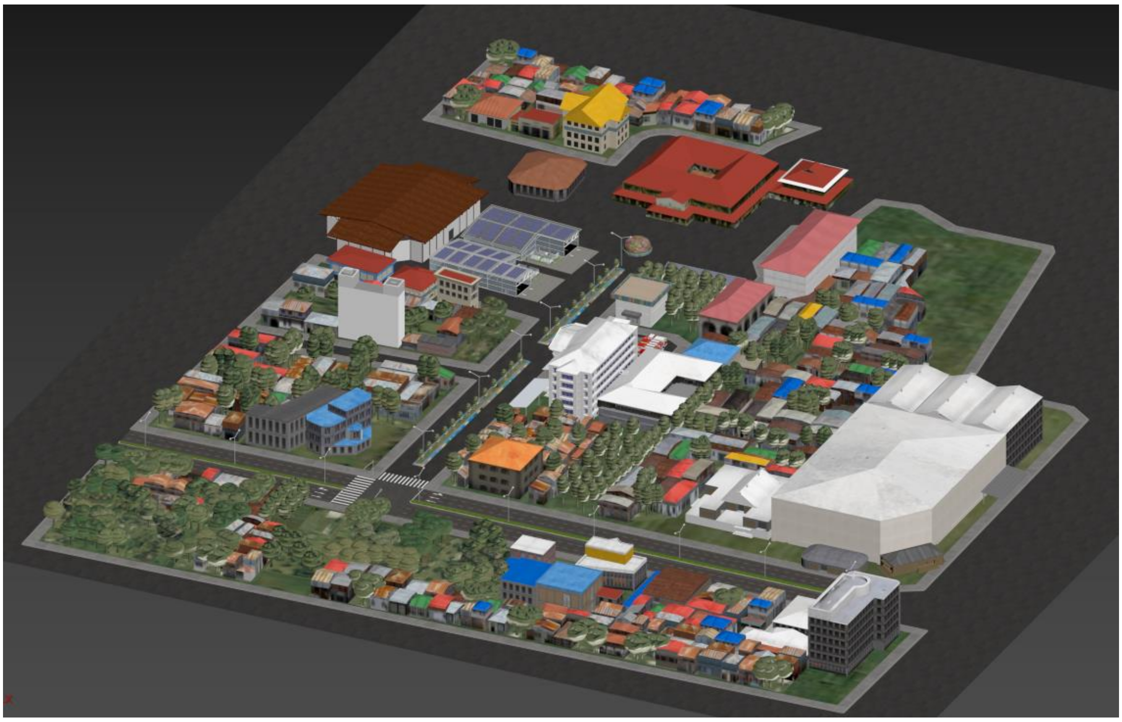


Figure 3.92 Urban Area Model in 3ds MAX

Finally, the urban area is modeled in 3ds MAX, this model can be exported to Unity in format .fbx

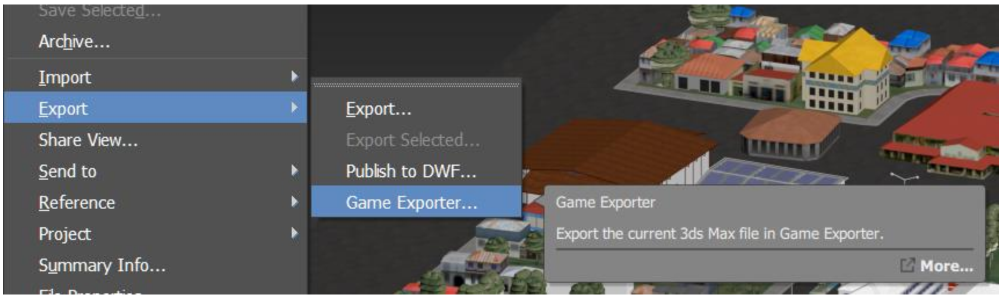


Figure 3.93 Game Exporter in 3ds MAX

The Game Exporter provides a streamlined workflow to export models and animations clips in FBX, and then we should import the FBX model into Unity.

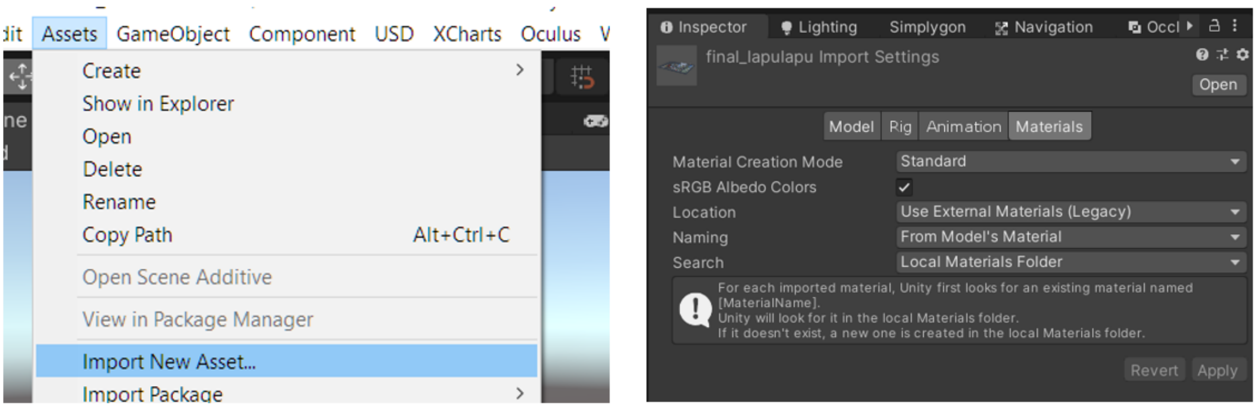


Figure 3.94 FBX Import and Texture Setting in Unity

In Unity, we can import the FBX model by select "import New Asset" in "Assets" menu, after importing, it may happen that the objects imported are not textured, that is because we have to configure the texture source of the objects, select the model, in "Materials" sub-panel of "Inspector" panel, we need select correct setting as shown in **Fig. 3.94**. Inevitably for large model which containing many objects, there will be texture loss for some objects, we should adjust it manually.

Figure 3.95 Urban Area in Unity and in Google Earth

As we can see from **Fig. 3.95**, the modeled area still looks different with respect to real world because the textures are not captured from the interested area, but the characteristic and the land use of the area can be reflected from the digital model.

Until now, the traffic environment has been prepared in Unity, in next section, the workflow of the traffic scene building through co-simulation between VISSIM and Unity will be described.

3.5.4 Traffic data exchange between VISSIM and Unity

The add-on module of VISSIM "Driving Simulator Interface" allows us to connect VISSIM to other software or devices, where VISSIM provides the surrounding traffic information such as vehicles position, speed, orientation, the traffic signal state etc. the driving simulator provides the data relative to the car driven by user or algorithm, when VISSIM received the driving simulator's vehicle data, this vehicle will appear in the road network of VISSIM as long as the coordinate received by VISSIM is positioned on a lane, since the data exchanged only include some number like the coordinate, speed etc. The road network in Unity and the road network in VISSIM has to be aligned lane by lane, them coordinate has to be the same, otherwise the vehicles position will never be correct.

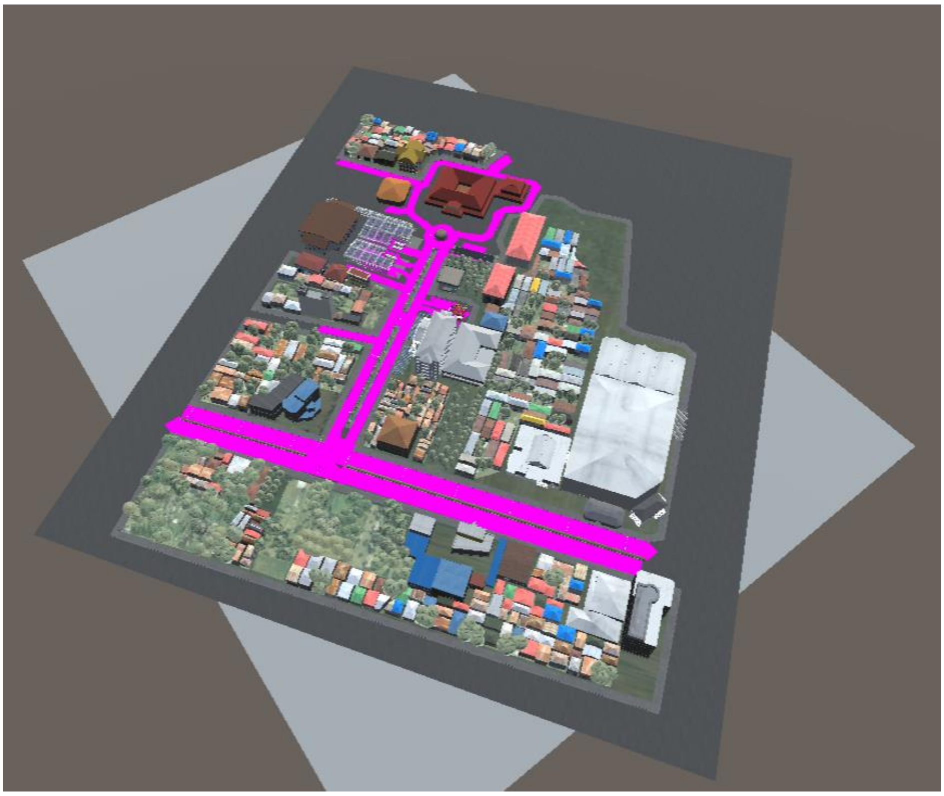


Figure 3.96 Road Network Alignment in Unity

Since the road network generated before (**Fig. 3.77**) is exactly same with the road network in VISSIM, we can use it as the reference to adjust our urban area model (**Fig. 3.96**), this step need to be done carefully since it is very important for the co-simulation, even if there is only one meters deviation, could causing strange behavior of the vehicles in Unity such as hit the wall, running on the curb, driving on the wrong lane etc. After alignment, the original roads (purple lanes in **Fig.**

3.96) need to be removed from the scene to reduce the impact on the performance due to the increasing of the tris and verts rendered by GPU.
To build the interconnection, Unity has to connect to a window DLL file provided by VISSIM named "DrivingSimulatorProxy.dll". Dynamic link library (abbreviated as DLL) is a program module that can be shared by other applications, which encapsulates some routines and resources that can be shared. In the "DrivingSimulatorProxy.dll", some program functions have been encapsulated which allow other programs such as Unity calling them in order to communicate with VISSIM through shared memory on the same computer. The functions used by us to build the interconnection will be explained bellow.

VISSIM_Connect: This function builds the connection between Unity and VISSIM, it will start VISSIM and passes some configuration data such as VISSIM version number, **.inpx** file name, simulator frequency etc. to VISSIM through shared memory. The return of this function is a bool value, true if the connection built successfully, else false.

VISSIM_Disconnect: Stop the simulation in VISSIM, disconnect the DLL with VISSIM, and close the VISSIM. Returns true if upon success.

VISSIM_SetDriverVehicles (VehicleNo, *VehicleData): Passes the data of vehicles controlled in Unity to VISSIM from DLL, the **VehicleNo** indicates how many vehicle's data should be transferred, if set to zero, then no vehicle data need to be transferred, **VehicleData** includes the driving vehicle's information such as position, orientation, speed etc.

VISSIM_DataReady: Returns true if VISSIM has already calculated all the traffic data for next time step, else return false.

VISSIM_GetTrafficVehicles: This function is called to get the vehicles data in VISSIM, the output data include vehicles number exist in VISSIM's road network and the relative data of these vehicles (position, ID, vehicle type, orientation, speed etc.), it will be blocked if **DataReady** returns false

VISSIM_GetSignalStates: This function is called to get the signal state in VISSIM, the output value includes the number of signal groups and the signal state for each of them.

The DLL is writing with C++, so the functions included are unmanaged type, to calling them in C#, the **DLLImport(" *.dll ")** function in namespace System.Runtime.InteropServices must be used. Additionally, the vehicle data are saved with struct in C#, which is managed object, the input and output data for the functions in DLL should be the pointer to an unmanaged block of memory, the methods **StructureToPtr** and **PtrToStructure** of **Marshal** class in namespace InteropServices should be used to make these data transformation. The scripts used to build the communication are modified based on "VissimInterface.cs" and "DrivingSimulatorInterface.cs" C# scripts in [87].

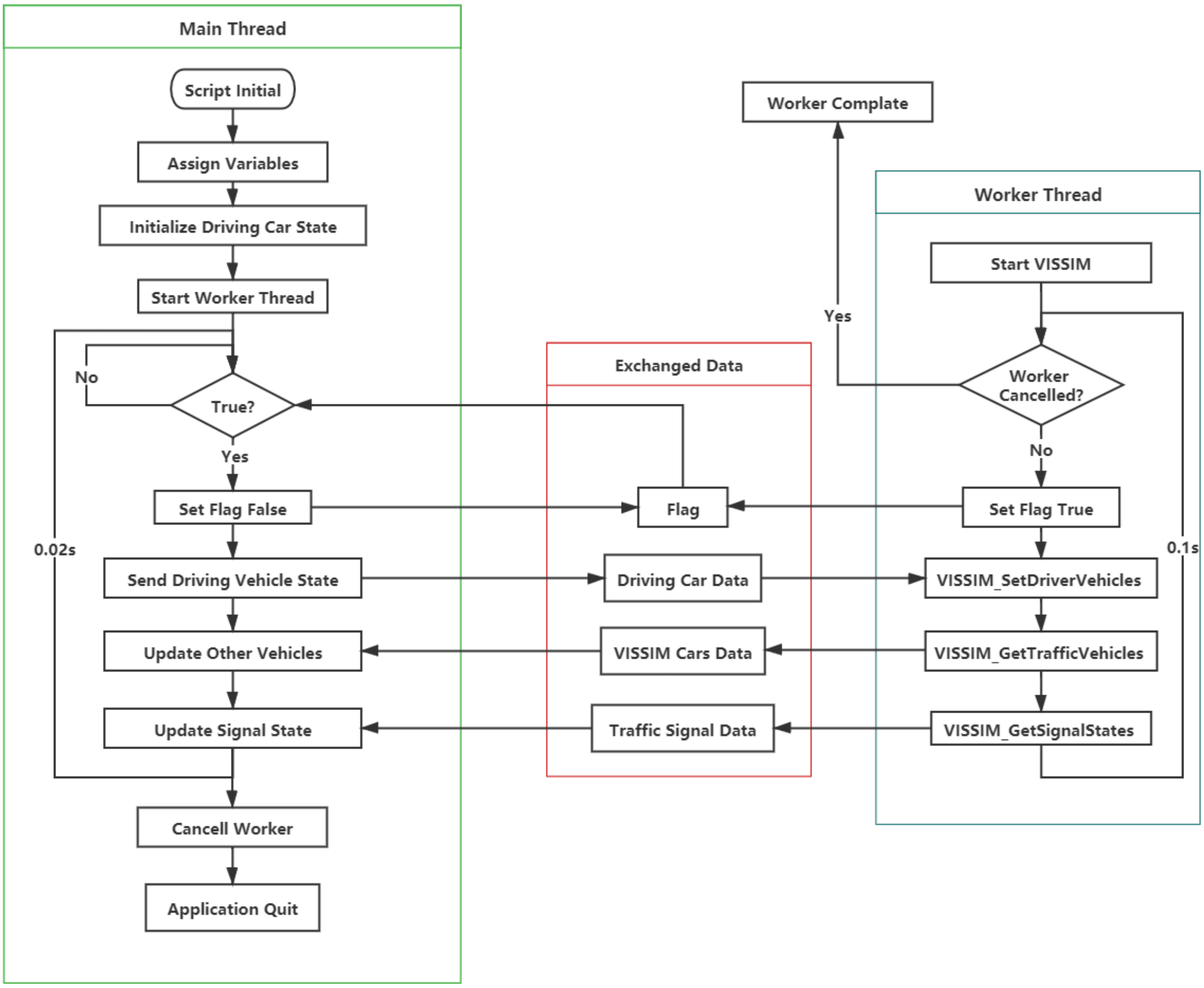


Figure 3.97 Script for Data Exchange

As shown in **Fig. 3.97**, all the DLL functions in "DrivingSimulatorProxy.dll" are called by the worker thread instead of main thread to decrease the influence on the performance, in main thread, the function "Start" will be called at initial of the program, in this function, the game objects and the prefabs need to be assigned to the variables, in order to spawn the corresponding vehicles when there are new vehicle data received from VISSIM. The function “FixedUpdate” will be called each 0.02 second by default, which is more frequent than the simulation step of the VISSIM, so if there are no updating of the data in “Exchanged Data”, the flag value will be set to “False” until the worker thread exchange the data with VISSIM and update it, in this period, the main thread will skip the current cycle and try again in next 0.02s. Once exchanged data updated, the flag becomes true, main thread will update the driving vehicle state data, read the updated data of other vehicles which perceived from VISSIM simulation from last simulation step.

Figure 3.98 Vehicle Prefabs in Unity

The vehicles data output from VISSIM is transferred into an array of struct, the struct not only includes the vehicles position, orientation, speed of last simulation step, but also specified the vehicle’s type, color, ID etc. Unity needs spawn the same type of vehicle in the scene, otherwise the vehicles in Unity will not behave appropriate, so, for each type of vehicle in VISSIM (vehicles used in microscopic traffic simulation), we have to prepare the vehicle prefabs in same type, they don’t have to be exactly the same, for example, the color of prefab in Unity can be different from the corresponding vehicle color in VISSIM, but the geometry property has to be the same, especially the vehicle length.

Every time received the updating of vehicle data from VISSIM in the FixedUpdate, our script should compare the car in the scene (a dictionary with vehicle ID as key and vehicle game object as value was created) with the vehicle ID in VISSIM (every car has its unique ID in VISSIM’s simulation), if the car in the scene included in new data list from VISSIM, then we should update its position, rotation, and speed using the newly received vehicle data, the speed apply to the vehicle’s rigid body can let the car move toward during the waiting time for next data exchanging (once per 0.1s, depending on the VISSIM's simulation time step). If the car ID in the new data list but we still do not have it in the scene, then we should spawn it in the scene and then add it in the dictionary which recording the existing vehicle in Unity. For the car in the Unity scene but its ID does not appear in the new data list from VISSIM, then we need destroy the corresponding vehicle game object.

The "Exchanged Data" in **Fig. 3.97** is a class used as the share data space between main thread and the worker thread, when one thread is reading and updating the Exchanged Data, it will lock the resources, to avoid another thread enter at the same time and causing error.

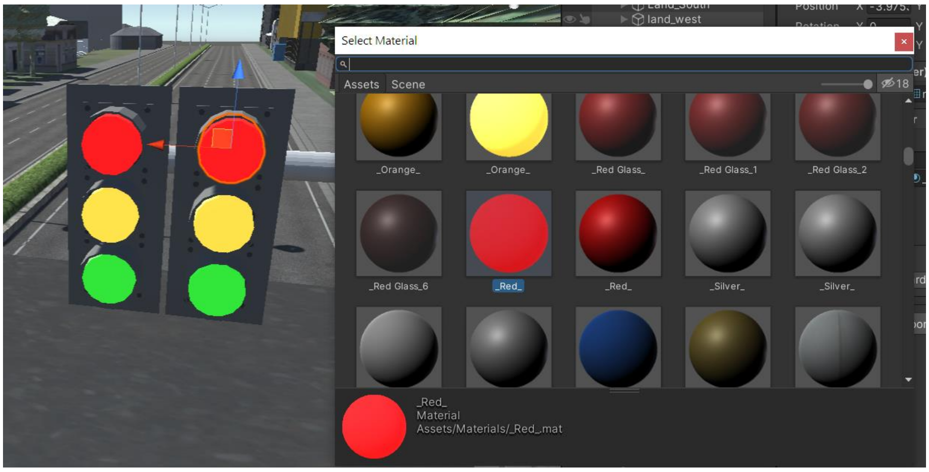


Figure 3.99 Traffic Signal in Unity

The realization of the traffic signal is similar, Unity receives the signal state of each signal group (the group EW_L, NS_L, and WE_S in **Fig. 3.51**) from VISSIM, then switch the models to the correspond material, for example, if a signal head's state is red, then we should switch the first cylinder to "_Red_" material, and other two cylinders switch to "_Black_". The EW_S's state should obtain from the bool logic of other groups as mentioned in **3.3.4 Traffic simulation setup**. The lighting materials are created by selecting "New Material" in right click menu, adjust the color to red/yellow/green, and then check the "Emission" selection to have the lighting effect.

Since in the digital twin data visualization management system we only need to visualize the traffic flow of the area, but in the driving simulator (will be described in **3.6 Driving Simulator**) with VR devices, we need drive the car in the scene and interact with other vehicle, two version of data exchange script will be used. In the first script, the data exchanging is unidirectional, Unity will only receive the traffic data from VISSIM and not send any data of driving vehicle (which is not exist in the scene). In another script for the driving simulator, the data exchanging is bidirectional, Unity will update data of the vehicle driving by us in the scene with VR devices to VISSIM by calling the function **VISSIM_SetDriverVehicles (VehicleNo, *VehicleData)** every 0.1 seconds (when the “flag” in exchanged data class becomes true), in order to update the state of the car in the VISSIM simulation, the other vehicles in the simulation will “see” our car and interact with us.

3.5.5 Traffic Scene Result

Figure 3.100 Traffic Scene of Interested Area

As the result of this section (**3.5 Traffic Scene and Virtual Environment Implementation**), the interested urban area with the traffic flow can be visualized freely with a roaming camera in the digital twin data visualization management system to have an overall view of the traffic situation (the workflow to create the roaming camera will be described in Fanshu’s part). The vehicles in the scene will interact with each other and also interact with the traffic signal (**Fig. 3.100**).

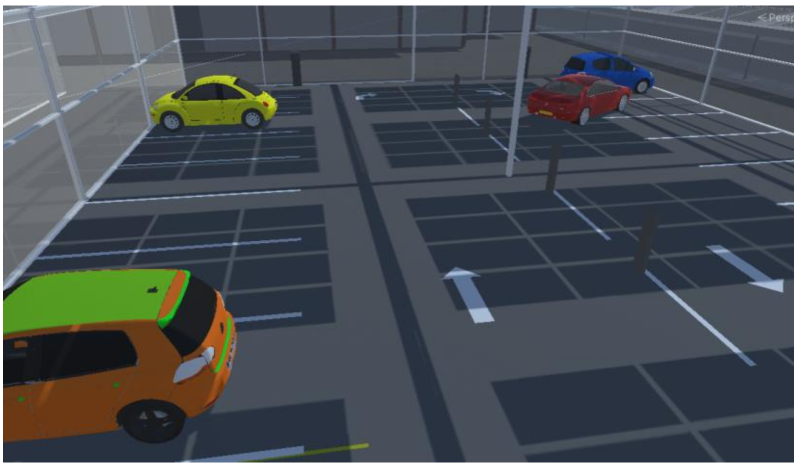


Figure 3.101 Parking Scene

Also, the vehicle accessibility of the parking space can be visualized and tested (**Fig. 3.101**).
The traffic scene is shown in Unity, but in fact, the behavior and state of the vehicle and traffic signal are calculated by VISSIM, then Unity receive these data by calling the functions above mentioned and showing them in its scene. Unity also provides the possibility to create the interaction relationship between its game objects (such as collision and sensor detection) and to apply the physical property to objects (such as gravity and friction). In next section, the workflow to create a driving simulator based on created traffic scene will be described.

3.6 Driving Simulator

In last section, the traffic scene based on the data exchanging between Unity and VISSIM has been built, to realize the VR interactive driving scene, the following elements have to be implemented.
Driving car: the vehicle game object which can receive the input from the devices like keyboard, joystick, then react and interact with the environment based on physic system in Unity and behave like the real car.
Interaction between driving car and other vehicles: the driving vehicle's data need be updated frame by frame to the traffic simulation in VISSIM, and then in VISSIM a car which has the same geometry property with the driving vehicle in Unity need to be created in road network, this vehicle does not follow the driving behavior model mentioned in **3.3.4 Traffic simulation setup** but change its state from the driving vehicle data updated from Unity.
VR connection: instead of driving in traffic scene using the computer's screen, the VR connection allows us to have a more immersive experience with its head mounted display (HMD) devices, where we can observe the surrounding environment by rotating our head like what is happening in real driving condition.

3.6.1 Driving vehicle setup in Unity

The vehicle moving is based on the friction between the wheel and ground. When driver press the acceleration pedal, the engine/electric motor of the vehicle will output a torque, this torque will transmit to the vehicle's drive wheel hub (depending on the type of the drive system which includes front wheel drive (FWD), rear wheel drive (RWD), or all-wheel drive (AWD)) through the dynamic transmission system (**Fig. 3.102**). Under this torque applied on the wheel hub, the vehicle's wheel will tend to rotate clockwise, so the bottom of the tire tends to move backward respect to the ground, to obstruct this moving tendency, the friction from the tire-ground contact patch direct forward will be applied on the vehicle, and this friction is the traction force which push the vehicle moving forward.

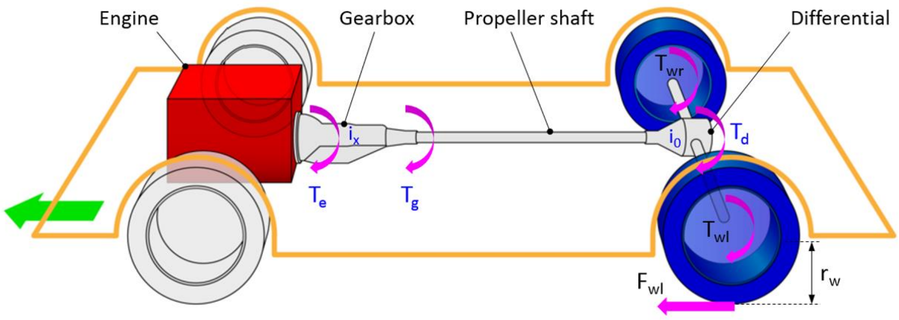


Figure 3.102 Vehicle Dynamic Transmission System

In Unity, to simulate the torque applied on the wheel hub and the friction between the tire and the ground, the “WheelCollider” can be used.

The wheel collider is a special collider to simulate the dynamic property of the vehicle wheels, it can simulate the spring and damper of the vehicle suspension system. It is controlled by the motor torque, brake torque, and steer angle properties, the collision detection of the wheel collider is done by emit the cast ray with downward direction with respect to the local reference frame of the wheel itself. [88]

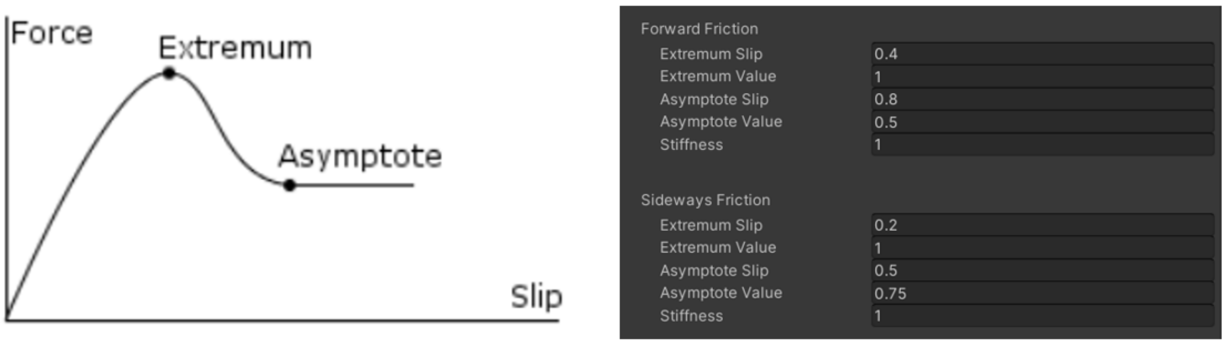


Figure 3.103 Friction Curve of the Wheel Collider

Wheel Collider uses a sliding-based friction model (left curve of **Fig. 3.103**) to calculate friction which is different from the normal friction calculation method of the physics engine. This friction characteristics is valid both for the longitudinal and lateral direction, the longitudinal slip reflects how big the difference between the real rotational speed and the rotational speed of free rolling of the wheel, where free rolling rotational speed can be calculated by dividing the vehicle current speed to the effective radius of the wheel. The side slip reflects how big the difference between the direction of the wheel and the direction of the wheel speed. The vehicle can move forward thanks to the traction force given by longitudinal slip and can turning thanks to the lateral force given by the side slip angle, but that does not mean higher slip, larger force can be achieved, like shown in **Fig. 3.103**, after the extremum point, the slip become to big that the tire start to sliding with respect to ground surface, and the friction force generated become lower than before, which enter into unstable region (Asymptote part).

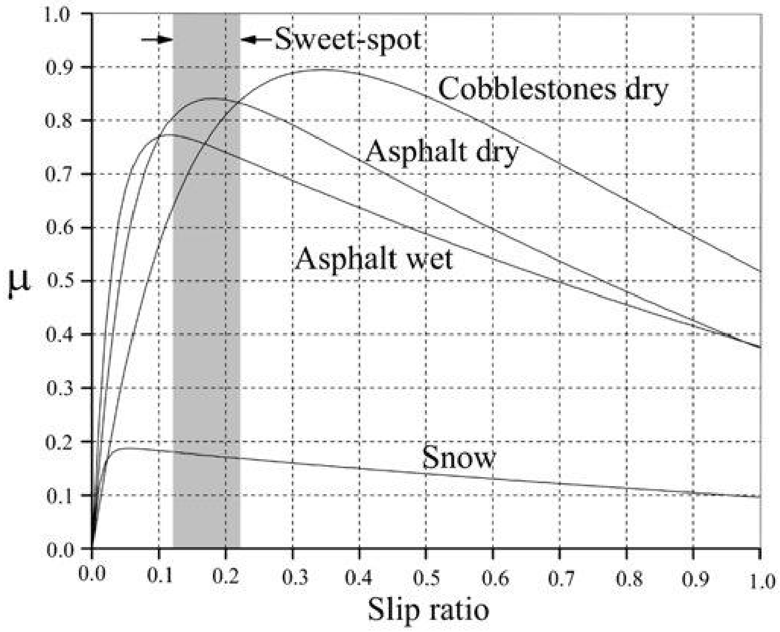


Figure 3.104 Friction Curve for Different Type of Roads [89]

This friction calculation method allows for more realistic behavior, but also causes the wheel collider to ignore the standard physic material setting, we can adjust the extremum and asymptote point by changing the parameter in **Fig. 3.103** to fit the friction coefficient curve of different type of road.

The other important properties of wheel collider include:

Mass (kg): the mass of the wheel which can be set in the wheel collider panel.

Radius (m): the radius of the wheel, should be adjusted to fit the size of the wheel model.

Suspension Distance (m): this item is used to set the maximum elongation distance of the wheel collider, which is calculated according to local coordinates. The suspension always extends downward through the Y axis of its local coordinates.

Center: should be adjusted to make the center of wheel collider coincide with the geometry center of the wheel.

Spring stiffness (N/m) and damping coefficient (Ns/m): higher the spring stiffness, larger the bounce speed, higher the damping coefficient, larger the damping force.
Target Position: the distance along its direction when the suspension is stationary. When its value is 0, the suspension is fully extended, when the value is 1, it is fully compressed.
Sprung Mass (kg): the mass of the vehicle body which supported by this wheel collider, its value should set in the "Rigid Body" component which belongs to the root game object of the wheel collider.
Force application point distance: application point of the suspension and tire forces measured from the base of the resting wheel, usually lower than the gravity center of the vehicle but higher than the ground level.

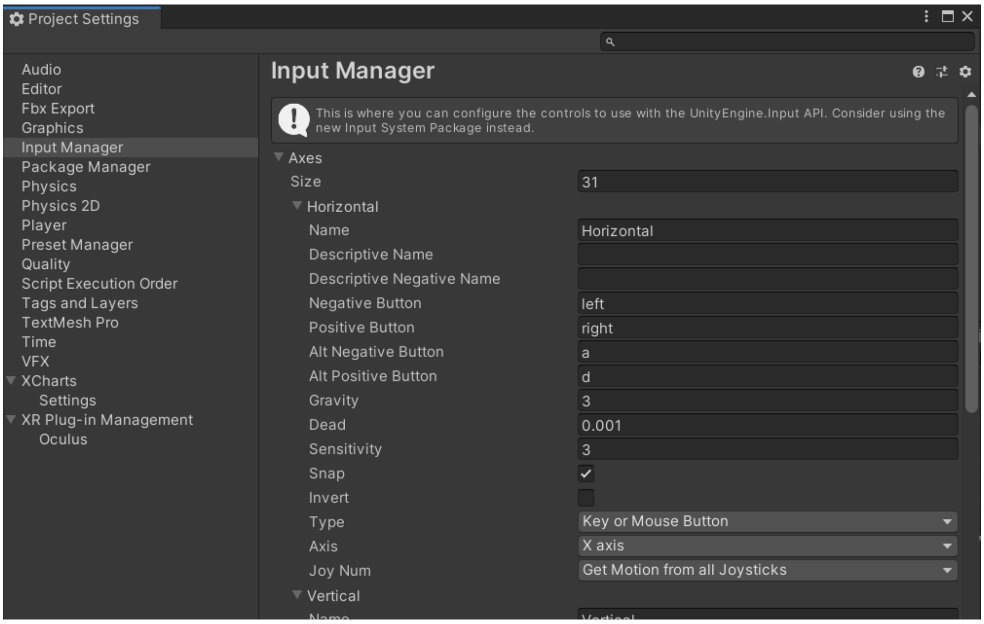


Figure 3.105 Input Manager

In the "Input Manager" section of "Project Setting" panel, we can configure our input method, and then in script, by calling the function **Input.GetAxis(Axis's name)**, the value between -1 to 1 will be returned depending on our input.
In the Unity scene, firstly we need create an empty game object as the root object of driving vehicle, then add a "Rigid Body" component to this game object, next we need to change the mass in the "Rigid Body" to a value supposed to have by a car (by default this value is only 1kg). Now drag the vehicle model into the empty element, and add the collider to it, the collider will let the vehicle body impenetrable by other objects with collider. Then create again the empty game object under the root empty model, and under this empty game object, we need create 4 empty game objects and

add the “wheel collider” component to them, after configured the different parameters above mentioned, the core elements for a drivable vehicle have been down.

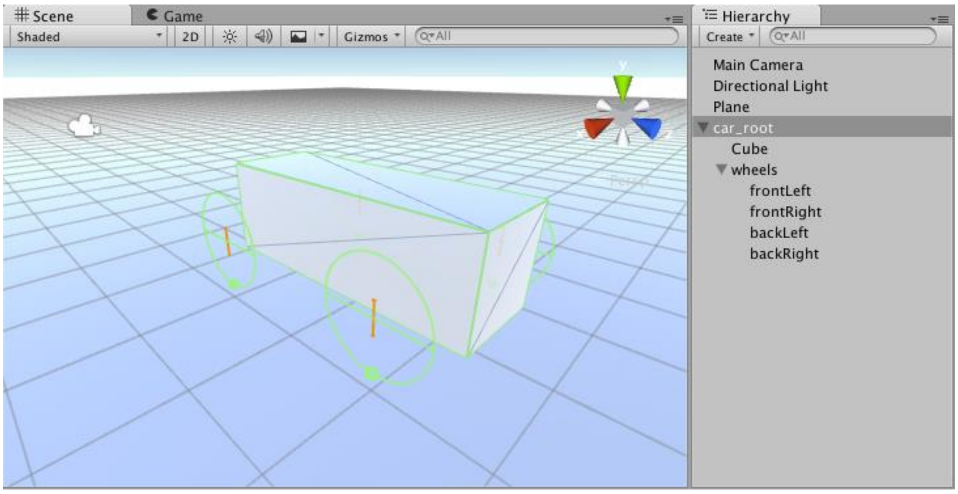


Figure 3.106 A Simply “Vehicle” from [90]

The vehicle model can be very simple like **Fig. 3.106**, with the help of wheel collider, we could get input from input system, apply torque to wheel collider by setting the “motorTorque”, apply steering angle by setting the “steerAngle”, then the physic system of Unity will calculate the slip of the wheel through the vehicle speed and the wheel rotational speed (the rotational speed is depending on the angular acceleration due to the torque balance around the wheel center), then the friction can be calculated and apply the force to whole vehicle object in the force application point configured before. This vehicle is ready to drive.

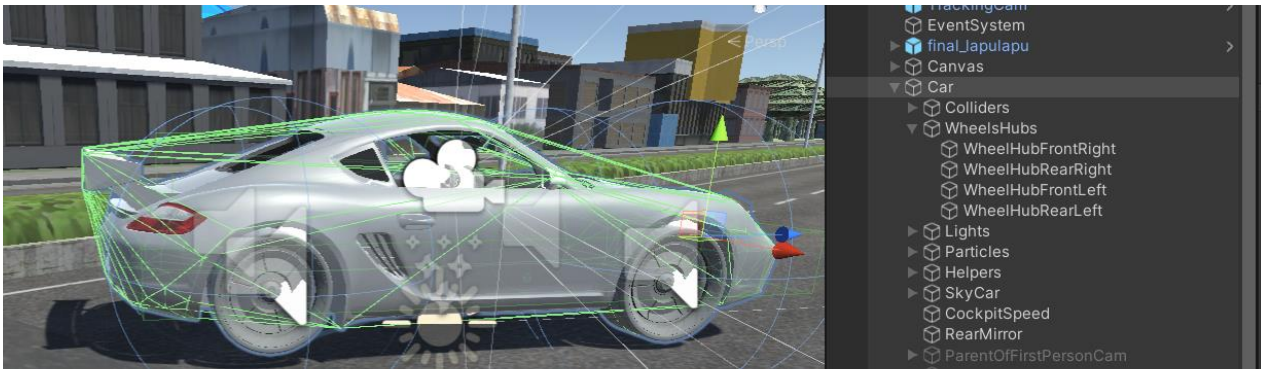


Figure 3.107 Vehicle Model from [87]

In our case, the vehicle model from [87] has been used because it has an interior model which is better for the first person driving with VR devices. In this case, we have also the model for the vehicle wheels, so the wheel collider should be adjusted to coincide with the wheel model.

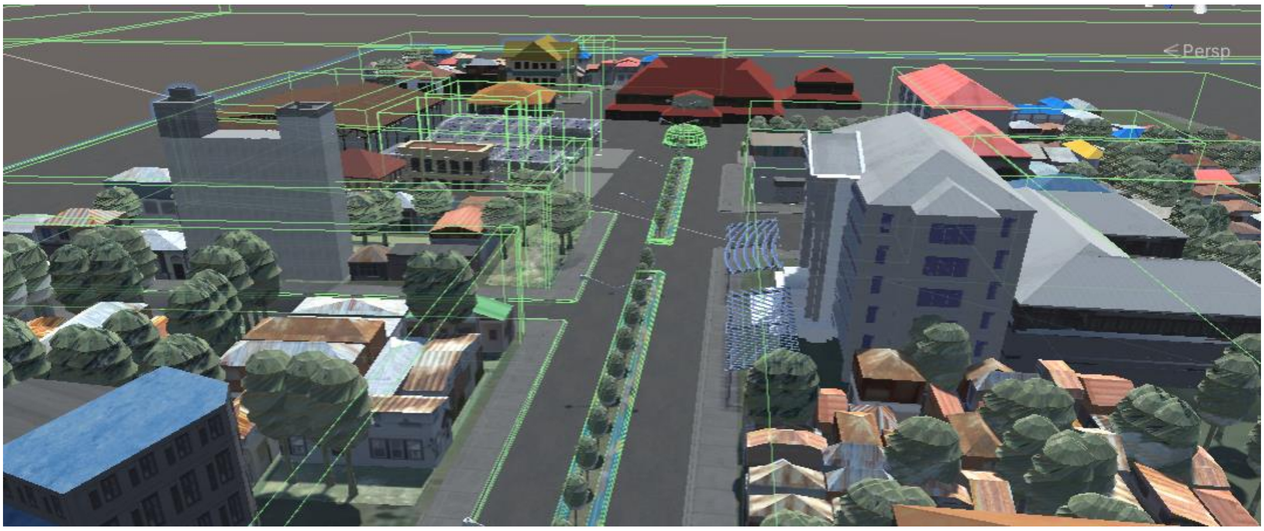


Figure 3.108 Collider for the Environment

The collider has to be added for the ground surface, because from the vehicle point of view, only the game object with collider is physical object, otherwise it will through the ground and falling down. The collider also added to buildings and the curbs to let the car interact with them and makes the scene more real.

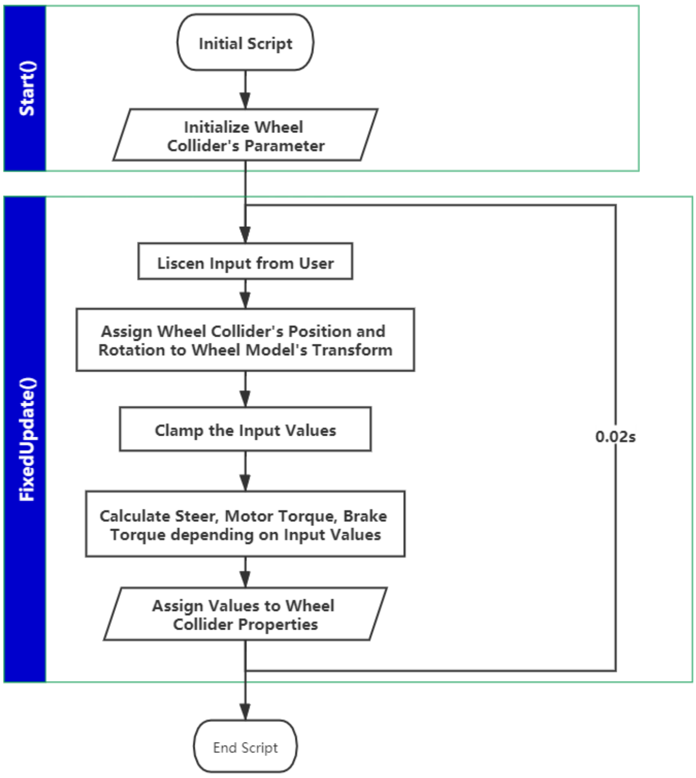


Figure 3.109 Script Workflow

The driving vehicle control script has a structure shown in **Fig. 3.109**, during the initialize parameter phase, the parameters configured like above mentioned will be read and assign to the properties of wheel collider, then listening user's inputs frame by frame, the input values after clamp is between -1 and 1, if the value is zero, that means there are no inputs for this axis. Next is to calculate the steer, torques based on input values and the properties of the wheel collider, for example, if the max steering angle is set to 25 degrees, the input value for steering axis is 0.5, then the steer angle for wheel collider is 12.5 degree. To have a better visual feeling, the wheel model should move and rotate with the wheel collider by updating its position and rotation, but we should know the dynamic property and torques is applied to the wheel collider, not the vehicle wheel model, the vehicle can drive even with no wheel model. Now in Unity we have the car ready to drive, the next step is building the data connection between driving car in Unity and the driving car in VISSIM.

3.6.2 Interaction between driving car and other vehicles

The driving car is controlled by people, the driving behavior of this car is depending on the people who operate it in the Unity, but if this vehicle is not updated correctly in VISSIM's traffic simulation road network, the other vehicles can not "see" the driving vehicle and has no interaction with it.

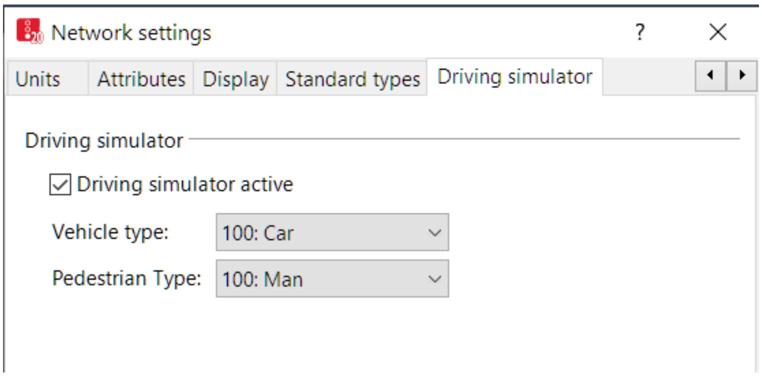


Figure 3.110 Network Setting in VISSIM

Firstly, the driving simulator mode has to be activated in Network settings panel in VISSIM, and then the Vehicle type of the driving vehicle has to be specified, this is the type of the vehicle which spawn in the VISSIM's road network and update its state from the driving vehicle data received from Unity.

In Unity, the script structure follows the flowchart described in **Fig. 3.97**, the state of driving vehicle has to be sent to VISSIM by calling the function **VISSIM_SetDriverVehicles (VehicleNo, *VehicleData)** each simulation step.

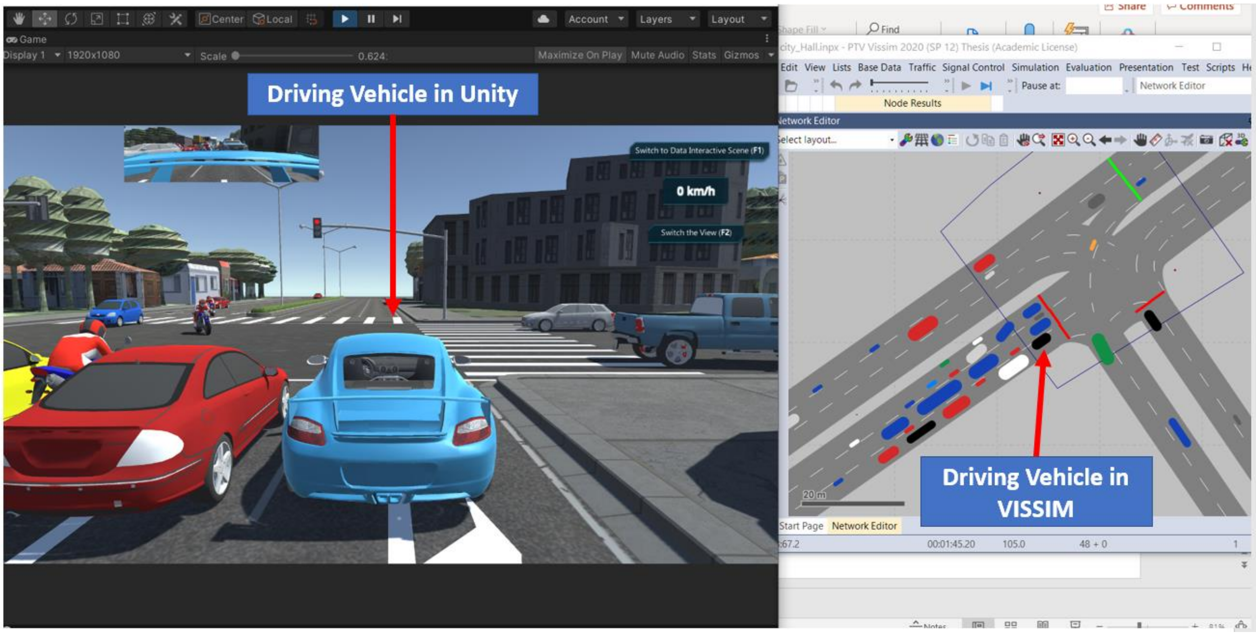


Figure 3.111 Data Exchange for Driving Vehicle

As shown in **Fig. 3.111**, when we are driving the car in Unity, the vehicle state data will transmit to VISSIM through C# script, and in VISSIM, the driving vehicle will update its state frame by frame in road network, the other vehicles in simulation will "see" us and interact with our vehicle.

3.6.3 VR Connection

Until now, our driving process is doing with the keyboard as the input method and computer screen as display devices, the plane screen does not feel like reality, with the help of VR devices, we can achieve an immersive driving experience.

The content showing in the "Game" panel after we click "Play" in Unity is the picture captured from camera, the "Camera" is a component in Unity, usually a empty game object with the "Camera" component form a camera in the scene, and when the scene start, the content showing in display is the figure captured by camera, in the inspection panel, we can configure the properties of camera where we can set the FOV axis (field of view axis), field of view (camera view angle respect to FOV axis), view port rect (indicate where on the screen this camera view will be drawn), clipping planes (the distances from camera where rendering starts and ends) etc.

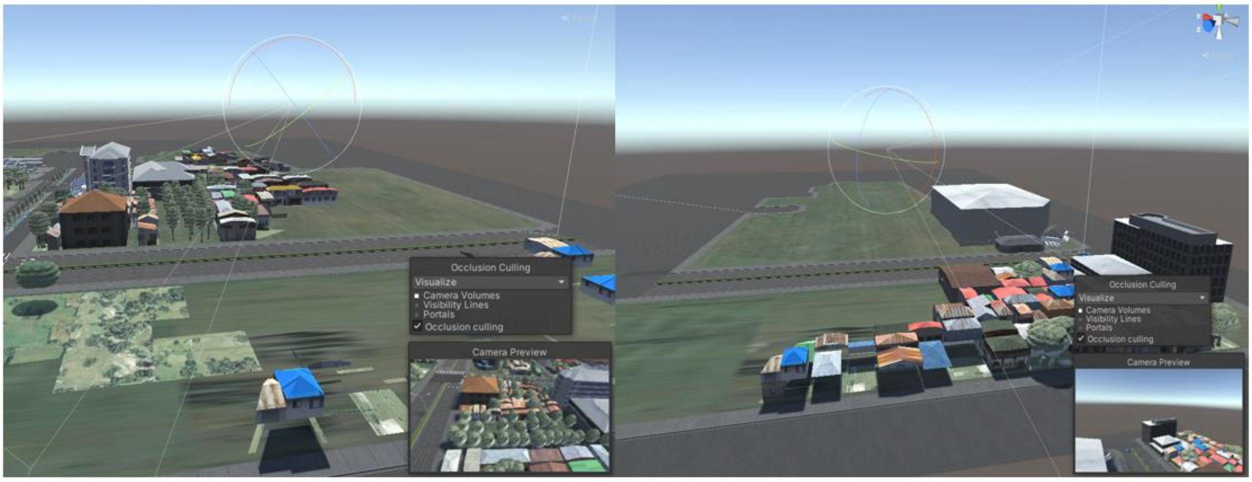


Figure 3.112 Occlusion Culling

Since what camera see is what we can see, GPU does not have to rendering every game object in the scene. Only render the objects inside of the camera view will reduce the stress on GPU and increase the running performance, this process named "Occlusion Culling" (**Fig. 3.112**).

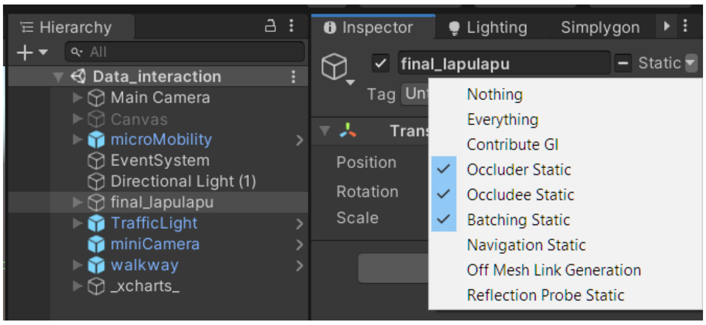


Figure 3.113 Static Batching

Additionally, rendering efficiency can be increased with "Static Batching", when Unity is building, it will automatically generate the merged grid and store the merged data in the form of a file, so that when the scene is loaded, the entire vertex data of merged model is submitted at one time, according to the engine's scene management system to determine the visibility of each sub-model. Then set the rendering state once and call multiple draw calls to draw each sub-model separately. The static batching can only be used for the game objects who does not move (such as roads, buildings etc.), and it can be activated by check the option in **Fig. 3.113**, the disadvantage is the file generated occupy the space of hard disk, the render efficiency increased because we trade "space" for "time".

Normally, with the computer screen, the tracking camera is set outside behind of the vehicle, because when we inside the vehicle, the view is too narrow that we have to rotate the camera to observing the surrounding environment but rotate the camera with mouse and keyboard is not so convenient, so put the camera behind outside the vehicle can obtain a wider view but lose the reality of the view port. In this case, something like view obstruct cannot be tested because we are not watching from driving pilot. This problem can be solved with VR device.

Figure 3.114 Oculus Rift

The VR device used by us is the Oculus Rift (**Fig. 3.114**), it includes a head mounted display (HMD), 2 controller named "Oculus Touch", and 2 sensors for detecting the "Oculus Touch" and the HMD by picking up the imperceptible glow of infrared lights all over the headset and controllers. The connection of Oculus Rift request 1 HDMI (High-Definition Multimedia Interface) output port and three USB (at least one 3.0) ports, then plug in the devices correspondingly, the application of Oculus will pop out and check if the connection is fine.

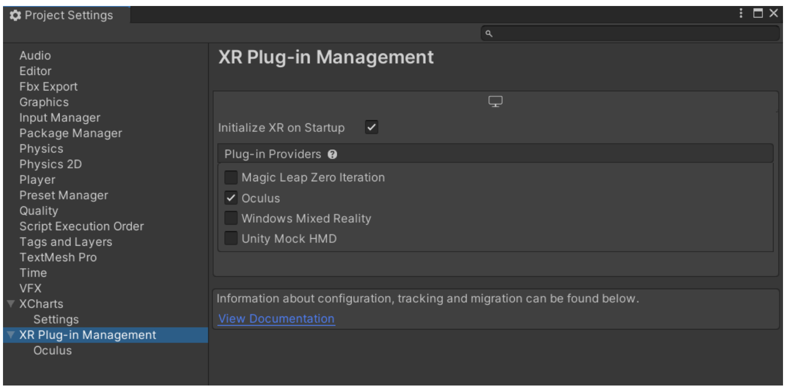


Figure 3.115 VR Setting

In Unity, the Oculus should be checked in "XR Plug-in Management" panel of the project setting (**Fig. 3.115**), in our case, the Oculus integration package is imported to realize the VR function. In this package, some prefabs are already made such as the "CameraRig", "ControllerPrefab", "HandPrefab" etc. since in the driving scene, we just need sit in the car and don't have to walking in the car, the "CameraRig" prefab is used.

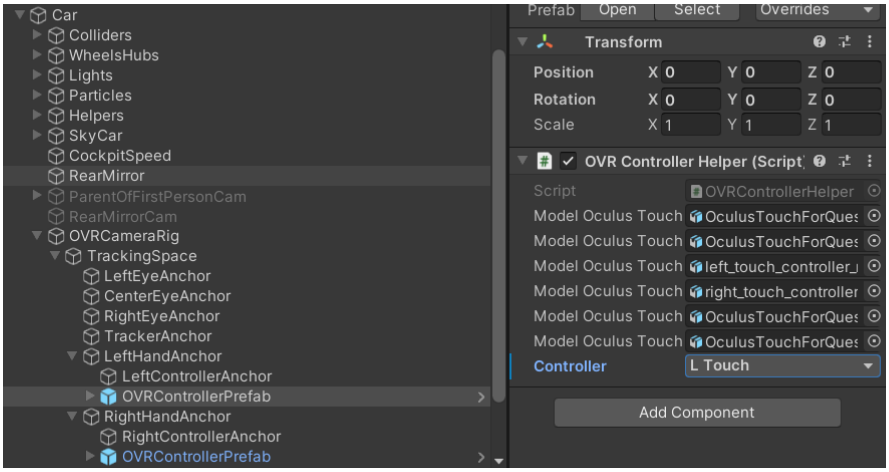


Figure 3.116 "CameraRig" Prefab

This prefab is already included some script as the component attached to its game object (**Fig. 3.116**), the type of controller needs to be specified so the script can read our input from "Oculus touch".

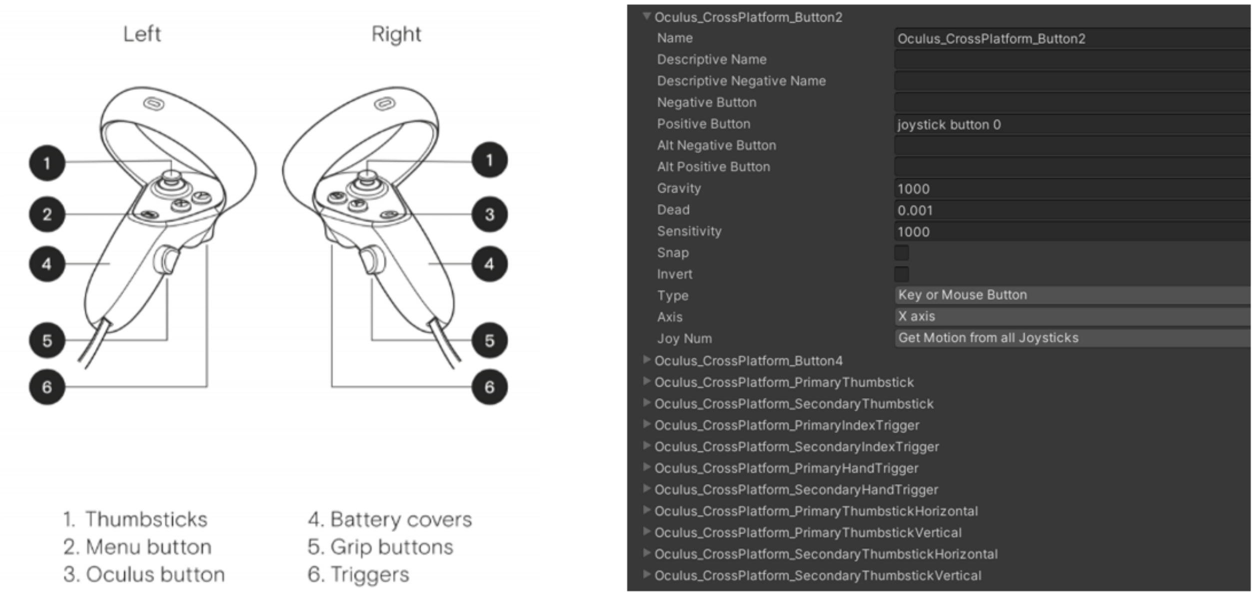


Figure 3.117 "Oculus touch" Inputs Buttons

In the "Input Manager" panel the several input axes appeared which correspond the buttons and triggers of "Oculus touch" controller (**Fig. 3.117**), to drive with this controller, the input values from different buttons and triggers should be listen by the vehicle controller script, and then apply the steer, torques to the wheel collider.

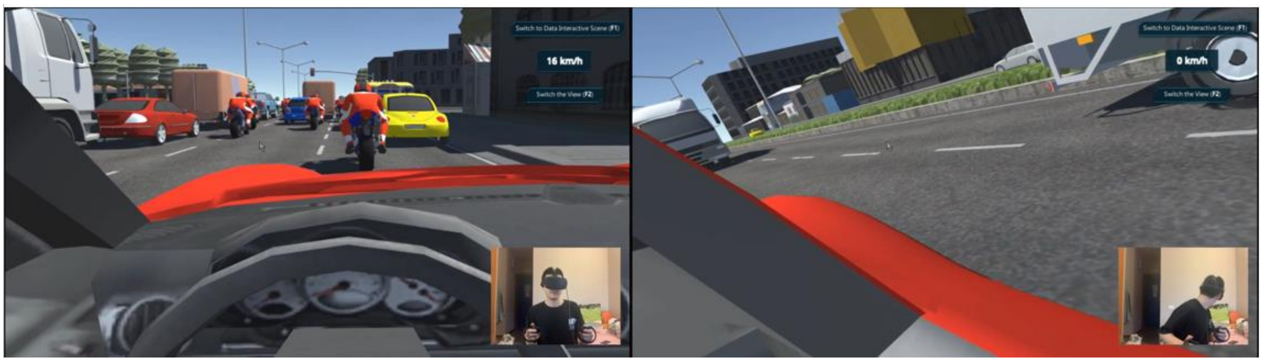


Figure 3.118 Driving with VR

Then the “CameraRig” prefab should be dragged under the car game object and adjust its position like the head of driver in the cockpit. As we can see from **Fig. 3.118**, the direction of the view can be changed with rotating of the head, which is exactly happening in reality.

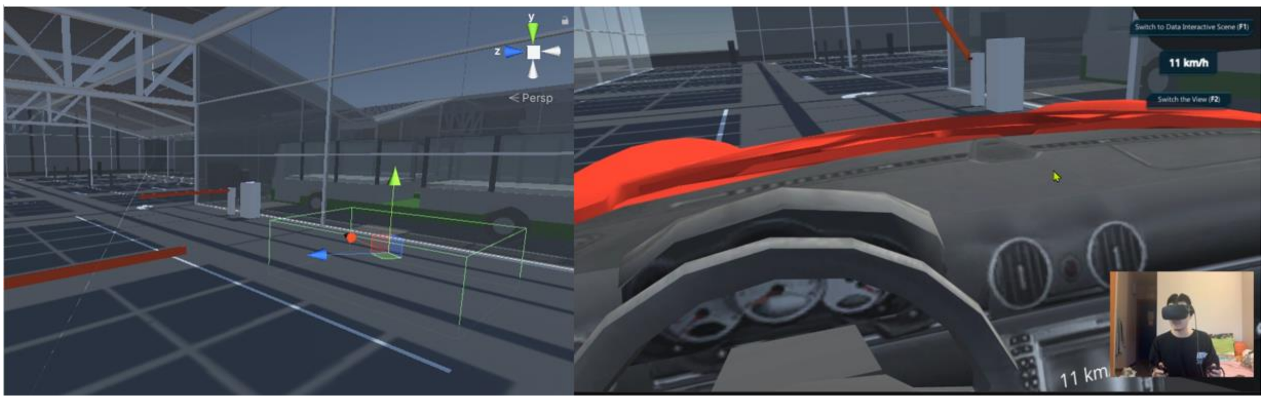


Figure 3.119 Sensor Simulated with Trigger

Additionally, the different type of sensor can be realized in Unity with camera, trigger, ray cast etc. For example, at the entry of the parking area, there usually has an inductive loop (mentioned in **3.3.3 Traffic data collection**) to detect the vehicle coming, in the Unity, we can use a collider and check the “is Trigger” option, this option will make the collider behave like the sensor and the vehicle can passing through it without collision. When other collider touches the area, the **Collider.OnTriggerEnter(Collider)** will be called, and the information of other collider will be passed into function as a parameter. With this method and doing the animation for the lift bar, when we are driving close to the entry of the parking hub, the bar lifts automatically to let us pass.

4. Conclusions and Future Development

4.1 Conclusions

In this thesis, relationship between traffic and environment pollution has been described, the traffic congestion increases the vehicle emission because it is causing the frequent stop and start of vehicle which makes engine of traditional car work at low efficiency and also makes the power consumption of the electric vehicle increased. Energy structure is a key factor in reducing pollution through electric vehicles, that means the development of renewable energy is very important for sustainable development realization.

Metro Cebu, as the second largest metropolitan area of Philippine, due to its fast urbanization recent years, the population growth fast, the development of road infrastructure and public transportation method cannot meet the traffic demand of its citizen, targeting to this problem, a smart mobility hub at city hall area and a moving walkway connect it to Mactan international airport was proposed to alleviate the traffic congestion and vehicle emission.

Digital twin, as an effective way to realize the smart city is also supposed to implement with the smart mobility hub area. In our case study project (formed by Daniele Iunti, Ma Fanshu, Zhang Zheyuan), a methodologic digital twin was implemented for the mobility hub area.

As a part of the project, this thesis has described a methodology to realize the traffic scene and driving simulator integrated in the digital twin management system which can be used for better management and visualization of the traffic, evaluate the accessibility of the parking hub, estimate the traffic capacity, test the different sensor solution etc.

Additionally, the ability of digital scene in Unity to change its state depending on the external data has been proven, which means as the increase of the amount of data, this digital scene will have more potential function can be developed and then generates more potential function to better service the reality world.

4.2 Future Development

Strictly speak, the traffic scene realized which described in this thesis still cannot be called as a digital twin of the traffic, because one important factor of digital twin system is the **Real-time** (mentioned in **2.2.1 Definitions and Characteristics of Digital Twins**) data exchanging from the real world. In our case, the microscopic traffic simulation has been used to simulate a real-world traffic and send the traffic data to our digital world and then realize the traffic scene, it did prove

the capability of our digital world to exchange data with external in real-time, but this real-time data source is still not the real world. The microscopic traffic simulation is built based on the data from the real world (the real traffic videos), but these data is historical data, not in real-time. So, for now, the workflow is using traffic simulator to transform the historical real-world data to the real-time non-real-world data and then transmit these data to our digital world.
In the future, as the IOT technology (various sensors and its communication) implemented in the area, the real-time and real-world data can be achieved to help the digital twin implementation, and more precise the data achieved, more function and services can be generated from the digital twin to makes the real world better. If we observe the data form received by digital scene, we can find out that the data needed by digital scene is the precise coordinate and angle of the vehicles, which is very hard to obtain from actual IOT technology, from **3.3.3 Traffic data collection** we can know there are several sensors can be used to estimate the real-time traffic flow data, these data can not be used directly to the traffic scene because it does not contain the precise state information for each vehicle, but we can use the real time traffic volume data to update the microscopic traffic simulation and then build the traffic scene through co-simulation, with this method, even if the traffic scene is not exactly same with the real traffic vehicle by vehicle, but since the simulation is based on the real-time traffic volume, the digital scene will be similar to the real world condition. So, in this phase, the flow is using traffic simulator to transform the real-time, real-world but not precise traffic data to real-time, precise, but non-real-world data and then transmit it to digital twin traffic system.
Another point can be improved for the methodology described in this thesis is the microscopic traffic simulation calibration based on interested area, as we know, the driving behavior can be very different from country to country, so to get a reliable simulation result, the calibration of the driving behavior based on the target region is necessary, this information can be achieved from the field inspection and measurement.
For the driving simulator part, the view experience is pretty realistic thanks to the head mounted display allows us to have a 360 degree view angle, but the input system does not feel so real because with the driving simulator mentioned in last chapter, the input system is formed by trigger and button, but in real world, we are driving with the steering maneuver and foot pedals, this problem can be solved easily by adopting the input devices for driving such as gaming steering wheel and pedals.
As above mentioned, the big challenge during the digital twin implementation is how can we get the real-time data precisely as possible and how should we process these data to make it useful for

us, the IOT technology and the big data analysis are the key solution for this problem. And as the IOT coverage areas increase, the urban digital twin area can be increased to increase its service coverage.

References

[1] Urban Population in the Philippines (Results of the 2015 Census of Population). Retrieved from https://www.statista.com/statistics/455910/urbanization-in-philippines/

[2] Cities and Municipalities Population Projections_2015CBPP

[3] List of cities proper by the population density. Retrieved from https://en.wikipedia.org/wiki/List_of_cities_proper_by_population_density

[4] JICA Project Team, 2019. Master Plan Study and Institutional Development on Urban Transport System in Metro Cebu

[5] https://en.wikipedia.org/wiki/Jeepney

[6] Niu GuoHui, 2015. Analysis and Research of Automobile Exhaust Pollutant

[7] L. R. Sassykova, Y. A. Aubakirov, S. Sendilvelan, Zh. Kh. Tashmukhambetova, M. F. Faizullaeva, K. Bhaskar, A. A. Batyrbayeva, R. G. Ryskaliyeva, B. B. Tyussyupova, A. A. Zhakupova and M. A. Sarybayev, 2018. The Main Components of Vehicle Exhaust Gases and Their Effective Catalytic Neutralization

[8] NOAA, National Centers For Environmental Information. Temperature Change and Carbon Dioxide Change

[9] REN Menglei, YANG Bin, LI Jun, July 2019. Greenhouse gases impact of electric vehicles considering the influence of energy structure and climate.

[10] ZHANG Lei, XU Ting, JIANG Daozhuo, LIANG Yiqiao, 2012. Comparative Analysis on Emission Between Electric and Conventional Vehicles.

[11] Carbon Brief, 2019. The lifecycle emissions figures were revised to reflect more recent data on electricity carbon intensity and battery manufacture.

[12] Vincent Aguiléra, Antoine Tordeux, 2011. integration of a speed-dependent emission model in dynamic traffic assignment: a large scale application to the Paris metropolitan area

[13] Sean D. Beeversa, DavidC. Carslaw, 2005. The impact of congestion charging on vehicle speed and its implications for assessing vehicle emissions.

[14] Wendell Cox, may/2003. How Higher Density Makes Traffic Worse

[15] Enjian Yao, Zhiqiang Yang, Yuanyuan Song, and Ting Zuo, 21 Dec 2013. Comparison of Electric Vehicle's Energy Consumption Factors for Different Road Types

[16] Andreas Braun, Wolfgang Rid, 2017. The influence of driving patterns on energy consumption in electric car driving and the role of regenerative braking

[17] Sun Zhicheng, Chen guang etc., November 2019. Effect of Actual Road Conditions on the Energy Consumption of Electric Vehicles

[18] Philippines Energy Situation. Retrieved from https://energypedia.info/wiki/Philippines_Energy_Situation#:~:text=The%20Philippines%20is%20located%20in,by%20wind%2C%20solar%20and%20biofuel.

[19] Installed generating capacity by source in Philippine. Retrieved from https://theaseanpost.com/article/why-philippines-so-focused-coal

[20] Philippines Declares Moratorium on New Coal Power Projects Amid Renewable Energy Push. Retrieved from https://www.terrenusenergy.com/philippines-declares-moratorium-on-new-coal-power-projects-amid-renewable-energy-push/

[21] Philippines Renewable Energy Policy Updates. Retrieved from https://www.egnret.ewg.apec.org/sites/default/files/geektic/files/%5BPhilippines%5D%20RE%20Law%20Policy%20Updates.pdf

[22] https://en.wikipedia.org/wiki/Mactan%E2%80%93Cebu_International_Airport

[23] Daniel Bell.The Coming of Post-Industrial Society: A Venture in Social Forecasting[M]. New York: Basic Books, 1973:212

[24] ZHANG Yuan-hao, ZENG Zhen-xiang, 2015. A Critical Literature Review of City Informatization -- from Cyber-Port, Digital City to Smart City

[25] Basil Blackwell. Manuel Castella.The Informational City: Information Technology, Economic Restructuring and the Urban-Regional Process[M].

[26] IBM Institute for Business Value, Smarter cities for smarter growth -- How cities can optimize their systems for the talent-based economy

[27] Apostolos P. Fournaris1, Konstantinos Lampropoulos1, Odysseas Koufopavlou, Electrical and Computer Engineering Dpt, University of Patras, Rion Campus, 2018. End Node Security and Trust vulnerabilities in the Smart City Infrastructure

[28] Y. Li, Yanliu Lin, S. Geertman, 2015. The development of smart cities in China

[29] Radovan Novotný, Radek Kuchta and Jaroslav Kadlec, 2014. Smart City Concept, Applications and Services

[30] Nasrin Khansari, Ali Mostashari and Mo Mansouri, 2013. Impacting Sustainable Behaviour and Planning in Smart City

[31] Cao XS, Yang WY, Huang XY, 2015. Accessibility and CO2 emissions from travel of smart transportation: theory and empirical studies[J]. Progress in Geography.

[32] Guo Jie, May 2020. Research on the Impact of Smart City Construction on the Improvement of Urban Traffic Congestion

[33] Kam-Yiu Lam1, Edward Chan1, Tei-Wei Kuo2, S.W Ng1 and Dick Hung. RETINA: a real-time traffic navigation system

[34] Wang Qiao, Li Ping, Lu Wei, Ding Xiaoyu, Lou Manrui, Huang Kaiming, Chen Shuo, Liu Xiaohu, He Peijun, Traffic flow information perception method based on contour and local characteristic point and system thereof

[35] Hua-pu Lu, Zhi-yuan Sun, and Wen-cong Qu, 2014. Big Data-Driven Based Real-Time Traffic Flow State Identification and Prediction

[36] Video Content Analysis retrieved from http://www.harzone.com/product/p_208.html

[37] An assessment of LTE-V2X (PC5) and 802.11p direct communications technologies for improved road safety in the EU. (http://5gaa.org/wp-content/uploads/2017/12/5GAA-Road-safety-FINAL2017-12-05.pdf)

[38] Marcin Seredynski, Bernabe Dorronsoro, Djamel Khadraoui, 2013. Comparison of Green Light Optimal Speed Advisory Approaches

[39] Grieves M W. Product lifecycle management: the new paradigm for enterprises. International Journal of Product Development,2005

[40] China Electronics Standardization Institute, November 2020. Digital Twin Application White Paper.

[41] YANG Lin-Yao, CHEN Si-Yuan, WANG Xiao, ZHANG Jun, WANG Cheng-Hong, November 2019. Digital Twins and Parallel Systems: State of the Art, Comparisons and Prospect.

[42] Glaessgen, Edward H. (NASA Langley Research Center Hampton, VA, United States), Stargel, D. S. (Air Force Office of Scientific Research Arlington, VA, United States), April 2012. The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles

[43] Colin Parris, Senior Vice President and Chief Technology Officer, GE Digital. What is a Digital Twin?

[44] Tao Fei, Liu Weiran etc. January 2018. Digital twin and its potential application exploration

[45] Aaron Parrott, Lane Warshaw. Industry 4.0 and the digital twin

[46] Siemens, Digital Twins: The Birth of Constant Optimization

[47] Grieves, M., 2015. Digital Twin: Manufacturing Excellence through Virtual Factory Replication

[48] Michael Grieves, 2021. Excerpt From Forthcoming Chapter Intelligent Digital Twins and the Development and Management of Complex Systems The "Digital Twin Exists ONLY After There Is A Physical Product" Fallacy

[49] Qinglin Qi, Fei Tao, Fei Tao, Tianliang Hu, Andrew Y C Nee, March 2021. Enabling technologies and tools for digital twin

[50] TAO Fei, LIU Weiran, ZHANG Meng etc. January 2019. Five-dimension digital twin model and its ten applications

[51] Sergey Ivanov, Kseniia Nikolskaia, Gleb Radchenko, Mikhail Zymbler etc. November 2020. Digital Twin of City: Concept Overview

[52] Li Deren, Yu Wenbo, Shao Zhenfeng, 2021. Smart city based on digital twins.

[53] Digital twin technology simulates the optimal solution of urban traffic, retrieved from http://www.xinhuanet.com/tech/2020-02/11/c_1125557837.htm

[54] Digital twin technology promotes the evolution of autonomous driving and smart transportation, retrieved from https://www.sohu.com/a/417673053_162522

[55] GE Yuming, WANG Yang, HAN Qingwen, 2019. Test Method of Connected and Automated Vehicles Based on Digital Twin

[56] Virtual, augmented, mixed: history and evolution of the new man-made realities. Retrieved from https://italiantechweek.org/en/news-en/virtuale-aumentata-mista-storia-ed-evoluzione-delle-nuove-realta-create-dalluomo/

[57] Zhang Liang, Jin Yi, Liu Yuanxia, Niu Li. Review of Virtual Reality (VR) Technology and Development

[58] Republic of the Philippines National Statistics Office Manila. 2007 Census of Population Report No. 2-56G Volume I Demographic and Housing Characteristics.

[59] Lapu-Lapu population density by barangays, retrieved from https://www.citypopulation.de/en/philippines/lapulapu/

[60] Ahmed Ould Sidibba, Diene Ndiaye, Menny O. El Bah, Sidi Bouhamady, January 2018. Analytical Modeling and Determination of the Characteristic Parameters of the Different Commercial Technologies of Photovoltaic Modules

[61] Zhanliang Tao, Jun Chen, 2015. Energy Storage Technologies in Smart Grid

[62] M Brackstone, M McDonald. Transportation Research Part F: Traffic Psychology and Behaviour

[63] Zhang Qinghua, June 2012. Calibration of VISSIM Microscopic Traffic Simulation Model

[64] Nurul Nasuha Nor Azlan, Munzilah Md Rohani, 2017. Overview Of Application Of Traffic Simulation Model

[65] Jinjun Tang, Fang Liu, Wenhui Zhang, Ruimin Ke, Yajie Zoue, 2018. Lane-changes prediction based on adaptive fuzzy neural network

[66] Ghaman R S,Zhang Li, Mchale G, et al. The role of traffic simulation in traffic signal control system development[J].

[67] Faghrp Ardeshir, Nanda Raman, Hamadc Khaled. Development of a dynamic traffic simulation model in a near system optimal route guidance system

[68] Richard bowling, Alexander Skabardonis, Vassili Alexiadis. Traffic Analysis Toolbox Volume III: Guidelines for Applying Traffic Microsimulation Modeling Software

[69] TSS. AIMSUN 6.0 Microsimulator User's Mannual.

[70] Caliper-Mapping Software. GIS, and Transportation Planning Software. http: www.caliper.com

[71] Sun Daniel(Jian), Zhang Lihui, Chen Fangxi, 2.13. Comparative study on simulation performances of CORSIM and VISSIM for urban street network.

[72] Arpan Mehar, Satish Chandra, S. Velmurugan, 2014. Highway capacity through vissim calibrated for mixed traffic conditions.

[73] Zhang Xiaodong, June 2016. DESIGN AND IMPLEMENTATION OF TRAFFICACCIDENT INFORMATION COLLECTION SYSTEM BASED ON VANET

[74] https://en.wikipedia.org/wiki/OpenStreetMap

[75] https://www.openstreetmap.org/export#map=17/10.31696/123.96555

[76] Key: highway in OSM. Retrieved from https://wiki.openstreetmap.org/wiki/Key:highway

[77] Video about the traffic of interested area. Retrieved from https://www.youtube.com/watch?v=0IALkp2306E, https://www.youtube.com/watch?v=F30ADMeisXI

[78] “highway” tag description in road classification system of OSM. Retrieved from https://wiki.openstreetmap.org/wiki/Key:highway

[79] VDOT VISSIM User GUIDE version 2.0, January 2020.

[80] PTV Vissim 2020 User Manual

[81] Bryan Higgs, Virginia Tech. Analysis of the Wiedemann Car Following Model over Different Speeds using Naturalistic Data

[82] Zhao Xiaosong, May 2017. Design and development of visual virtual simulation Experiment platform based on Unity3D

[83] Microsoft, August 2014. Unity : Developing Your First Game with Unity and C#. Retrieved from https://docs.microsoft.com/en-us/archive/msdn-magazine/2014/august/unity-developing-your-first-game-with-unity-and-csharp

[84] Jojokzhang, 2018. Introduction to Unity Engine and C# Script. Retrieved from https://mp.weixin.qq.com/s?__biz=MzI3NTM1MjExMg==&mid=2247484733&idx=1&sn=9c756898d809b940771e2649257b94c9&chksm=eb075a62dc70d3746bcdaf8c484167acf2746acd518eb9fd2b50f394d3103852e79fc4632989&scene=27#wechat_redirect

[85] FBX https://en.wikipedia.org/wiki/FBX

[86] Huang Liangping, January 2015. Research and Application of 3D City Model Method Based on CityEngine

[87] PTV VISSIM - UNITY DRIVING SIMULATOR Documentation

[88] Unity Documentation: Wheel Collider. Retrieved from https://docs.unity3d.com/ScriptReference/WheelCollider.html

[89] Tarun KUMAR Bera, K. Bhattacharya, Arun Kumar Samantaray, November 2011. Evaluation of antilock braking system with an integrated model of full vehicle system dynamics

[90] Wheel Collider Tutorial. Retrieved from https://docs.unity3d.com/2018.4/Documentation/Manual/WheelColliderTutorial.html