Investigation into 3D road infrastructure modelling for a bicycle safety study

Machiel van der Stelt
Student ID: 22888094

Acknowledgments

I duly thank Professor Thomas Braunl of the Department of Electrical, Electronic and Computer Engineering of the University of Western Australia for his advice, guidance and support in the ups and downs of student life.

I also thank Professor Lynn Meuleners and Dr. Michelle Fraser from the Faculty of Public Health at the University of Western Australia for their time, patience, and trust.

I thank Jesse Helliwell for his assistance and generosity in the beginning of this project. Jesse volunteered to visit me and explain about the workings of the electronics and software parts, as well as the working in general of this road safety bicycle system.

Also thanks to Einar Fridjonsson for his prompt and valuable answers and advice on my general questions in regards to this research project and paper work submission processes.

Finally, I thank my family and friends for their interest and support.

Abstract

The Road Safety Research, of the School Population and Global Health at The University of Western Australia is investigating a number potential safety measures such as road treatment and wider bicycle lanes to determine if any of these measures improve the road safety for bicyclists. The mode of the study is a modified bicycle with a number of sensors, which are connected to a computer with the use of 3D goggles for experiencing a 3D environment by the test subjects during the study.
The requirements of the 3D environment were that the environment needed to represent closely two locations in the Perth Metropolitan Area in Western Australia, particularly the road lay-out and the road signs
The purpose of this dissertation study is contributing to the Road Safety Study by researching and implementing a 3D virtual environment according to requirements mentioned for the Road Safety Research study. Additionally, providing a deeper insight into road safety technologies and the development of road safety technologies over the last 50 years will be provided.

This dissertation found that using RoadRunner modelling software and Unity3D game environment software provide an effective choice for creating a 3D environment for road safety studies. For instance RoadRunner is specifically designed to model road infrastructure and traffic, while Unity3D has a Game Engine and the ability to have equipment connected to it with sensors, such as joysticks.
There needs to be taken into account that virtual 3D environments have some limitations, such as drivers not being able to correctly determine the distance between themselves and objects within the 3D virtual environment, and maintaining the correct speed. Because a virtual 3D environment doesn't represent reality for 100%, and the lack of physical feedback to cyclists, findings in research studies involving a 3D virtual environment cannot be fully applied to reality.

Previously, research in Western Australia was conducted using hard bicycle crash data, the findings of this research were that crashes didn't involve major vehicles, but involved the bicycle crashing with surrounding objects. The Bicycle Safety Research study was proposed to investigate the bicycle crash situation in a safe environment. The method of this thesis research was to do a literature review, to have a better understanding of existing technologies for diverse testing settings..Also the best suitable software needed to be chosen, in addition to selecting the most appropriate spatial data. This thesis research has offered the best cost effective solution to simulate Bicycle Road Safety locations for collecting detailed data.
For future studies these considerations could be taken into account for a more realistic virtual 3D environmental experience; future studies for instance could consider to

additions, such as the adjustment of the test vehicle or bicycle, so that this object will be more realistic in terms of physical movement, or possibly the adjustment so the test object is better integrated with the 3D environment, so other parameters such as wind, etc. could be introduced into the 3D environment and experienced during the study.

Contents

1 Introduction

As roads are more crowded with less space to share, which may put a strain on road capacity and possibly puts the lives of vulnerable road users, like bicyclists, in danger.

To better understand the potential hazards to bicyclists on roads, studies are being developed to measure these hazards accurately. While performing these road safety studies, it is important not to endanger the lives of the test subjects. For this reason testing in a simulated and virtual environment is a good choice to fulfill this requirement.

As the virtual environment is not the real environment, it is important to simulate the reality as best as possible, with the current technologies and limitations that come with them.

This research will address the question if software and hardware technologies, which have come to a stage where they are very sophisticated, could be used for bicycle road safety studies to recreate 3D environments. Additionally this investigation will try to find the answer if the researchers can collect accurate, detailed and diverse data in a bicycle road safety research setting. Finally this research will address the added advantages and disadvantages of using a virtual environment, compared to real life situations for bicycle road safety studies.

In chapter 2, A literature review will be covered in relation to virtual 3D road environment creation as well as a number road safety studies. The main goal in this chapter is to understand how the software and hardware technologies have evolved over the years. For instance the oldest study reviewed in this chapter is from 1975 with a physical car simulation, while one of the newer studies is from 2018, in which a 3D virtual environment for bicyclists was investigated.

In chapter 3, Will outline the progress made in recreating a virtual environment with two software packages, RoadRunner and Unity3D. The required virtual 3D environment needed to be designed in a way that it represents two real life locations in the Perth metropolitan area in Western Australia. Also the 3D environment needed to have safety implementations, as used in Victoria, so researchers can better understand if these safety implementations have the same safety benefit in Western Australia. The two locations of interest for this study are a location with a roundabout, while the second location was a road with a long and wide curve. These two locations needed to be built into two scenarios for the study, while one scenario would the current situation without road treatment, the scenario would have with road treatment, such as signs on the pavement and adjusted widths of bicycle lanes.
This chapter will also discuss some existing technologies, which enable users to scan objects, by making pictures taken in a circle around the object of interest, and determine their suitability for 3D modelling road infrastructures.
Additionally this chapter will briefly discuss Object Oriented programming and how that could help to recreate realistic environments, while keeping the programming code to a minimum.

The following chapters will outline if 3D virtual environments, created with Roadrunner and Unity3D, are representative of the real environments and can be adequately used in Bicycle Road Safety studies to collect accurate and diverse data.

2 Literature review

A validation study [7] conducted by Steve O'Hern, Jennie Oxley, Mark Stevenson studied and compared the difference of bicycle driving behaviour on a road situation and compared the difference of the bicycle driving behaviour on a road situation and on a bicycle simulator. This study, as it already says, was designed this way to validate the bicycle simulation validity as compared to on road situations. The finding from the report is that there was some sort of validation of the bicycle simulator situation, but some notable limitations, as compared to on-road situations, such as speed awareness or experience and possible disorientation with head movements.

For studies involving detrimental outcomes, researchers need for ethical reasons, mostly use a Retrospective study, which means the researchers look at exposure-historical data (in this case characteristics of road infrastructure). For instance if researchers asked bicyclists to ride on roads where the researchers expect a crash with a possible hospitalization, that would be unacceptable for ethical reasons.

For this reason the study [6] by Meuleners L, Fraser M in 2018 was designed following the Case-Control model. This study set out to find an association between the characteristics of road infrastructure (exposure) and hospitalizations of bicyclists caused by crashes (outcome). The characteristics of a Case-Control study are that researchers look at a specific outcome or combination of outcomes, like a disease, injury of a certain bicycle population and compare that with a same size same kind of population (sometimes called a control group or compare group) who didn't have the outcome (in this case crashes leading to bicyclist hospitalizations).

One important finding from this study was that many of the bicycle crashes happened without the involvement of a major vehicle, but more due to surroundings as road conditions and objects on the road. This did lead to the recommendation that further research should be conducted into finding out the impact of different road conditions or treatments on the outcome of crashes; this research question is investigated in this new bicycle safety study.

In a 1975 study [15] the effect of Alcohol was assessed on the driving performance in a car simulator. While during this period the technology compared to current technologies available where very limited. It can be pointed out that the techniques with limited technologies in this study were impressive. For instance they used a CRT display of approximately 20 cm by 25 cm, which was mounted on the hood of an old car on the driver's side of the car. On the screen one would see part of the hood projected as well as the left side of the windshield outline.
As far as the road outline and scenario, lines were used to represent the left and right edges of the road and the scenario was presented as if driving in the night on a single lane road.
For the simulation itself the driver had the task to follow the road, which was mainly straight, while the steering wheel drove the monitor in two degrees of freedom to represent the steering movement in changes in the hood direction, vertically and horizontally.

The researchers implemented a simulation of wind gusts, for which the driver had to adjust the steering wheel to maintain the correct location on the road, while the speed was set to be constantly 30mi/h or about 48km/h. The driver's response was measured by correlating the control of steering with lateral deviations in addition to the heading of the car. In addition to this indicator lights and road signs were simulated. Two indicator lights were used by installing them on either side of the CRT display, and were used to indicate for the driver to either apply the brakes or use the horn. While another set of indicator lights were installed mounting them on the rearview mirrors, which were used to simulate road signs.
As this study is graphically limited compared to more advanced technologies used in current studies, it is clear that results are limited in terms of details and diversity. This study for instance was only able to measure the response of the drivers on events or changes to the road and the driving direction of the car. An approach to handle the limitations of this study, the researchers also scanned the eye and head movements to get a clearer idea how the drivers reacted to changes. Furthermore, the study concluded that drivers under alcohol tended to respond slower to changes and events and that they also were more likely to way of the course.

In a study [17] from 1994 describes the method of constructing a virtual environment, specifically for simulating driving through cities. The study mentioned that they used a high-performance workstation, which was a [21] IRIS CRIMSON workstation.
This workstation used either a 100 or 150Mhz microprocessor, and had a working memory of 256 MB, while the software was developed at a University. This study also argued that the high-performance workstation needed to be used in order to generate the virtual scenery in real time. Further to note is that the software was designed to simulate driving in an environment in addition to the ability to construct virtual towns and roads. The researchers describe that the software is capable of scanning current road infrastructure from a map, from where an 3D object can be generated.
The process of scanning map data can be divided into 5 steps. The first step is that a map image with roads on it is scanned. Then in the next step the begin point on the map is determined, after an edge on the map is specified. At step 3 the distance is calculated between the edge pint which was already inputted and the road which was already extracted and which is nearest. In step 4, step 2 and 3 are repeated for another end point on the road and an edge. In the final step, in step 5, the width of the road is represented by the distance between a number of specified points. It is quite a laboress process, but it appears that this process in 1994 was the best method to get map data as accurately as possible into a computer.
It appears that when one attempts to connect roads in this software, the roads don't snap automatically together, but that they manually need to be put together and that for this a number of rules need to be followed inorder to make the design correct. An example of the rules are for instance when two roads need to be connected. The two end points of the roads need to overlap each other and not overshoot to make sure that the crossing has a natural shape. The software also applies an automatic technique where the edges are moved towards the centre to promote the natural shape of the crossing.
A study [18] from 1999 describes the use of Object Oriented software in which different objects in a 3D environment can be programmed independently from each other. Also the ability exists that these programmed 3D objects can interact with each in a 3D software environment, with

traditional software packages or languages. The research paper names a number of available object oriented software languages, such Delphi, Visual C++ and Visual basic among others. The researchers did choose to use Visual Basic, because their study could then be adopted to simulate a wide variety of objects that could be packaged in class modules. Also, according to the researchers, there was no need to understand the inner workings of the models, when they were called on to be used in the software program.
In their conclusion the researchers claim that Object Oriented programming, and particular Visual Basic greatly simplify vehicle systems simulation, as well as in multi vehicle system simulations.

Another study [19] analyzed the difference of reaction time in hazardous situations between novice drivers and experienced drivers in China. This study was performed in 2007, when traffic in China was well and truly growing for a number of years. This fast growth resulted in a number of risk factors the researchers identified compared to westefrn nations. These risk factors were: first a larger proportion of novice drivers, second the quality of roads differ greatly in the infrastructure, from new highways to unpaved roads and the third risk factor was identified as that traffic consists of a large mix of motorized vehicles, bicycles and pedestrians. The researchers selected five different cases of situations which regularly happen in China, which they wanted to use in their simulated study.
For this study, like the study from 1975 discussed earlier in this paper, the cab of a real car was used. In front of the car was a cylindrical screen installed, with two projector, projecting the scenery on the screen. While this system is in terms of performance is an improvement to previous studies. It lacks one important capability, that is that the driver in the car cannot be simulated inside the software environment, as a physical cab of a car is used and is separate from the virtual 3D environment. For this reason no measurements can be performed in terms of the proximity of the driver in the car to other objects in the scenery. Thus study, however, like the study from 1975, measured the reaction or response time of different kinds of drivers (experience and novice) in different road situations. The findings from this study suggested that novice drivers have a less appropriate reaction to hazardous situations compared to experienced drivers. The researchers also noted that because this study was a simulation, and that for some situations which occurred in the simulation not necessary would happen in reality, such as accidents. The researchers further recommend that novice drivers need to have extra training with a simulator in learning to appropriately react to hazardous situations.

A literature study [12] from 2009 tried to answer the question if a driving simulator can be used to predict the risk of crashes in the real world. In this literature research many different simulation methods are investigated such as a car placed in front of a screen or a capsule where test subjects are placed in and which moves according to the situation and actions of the driver and the different kinds of actions of the driver. The literature review concludes that although simulation is reasonably representative compared to the driving on the road and its demands in respect to the drivers behaviour during driving. Further the researchers note that driving in a simulator is a first good step in evaluation situations, which were not tested before, additionally they note that it is a cost-effective, safe way of testing driving.

In 2012 a study [20] was performed in assessing the influence of in car music on driving through the use of a driving simulator. While this study was not from that long ago, in relation to the other studies, described, it is surprising that the technology they used was relatively simple. The system consisted of a motor race video game, a single small screen in front of the driver, while the driver sits in a racing chair with a steering wheel and pedal system in front of them. Interestingly this research had three methods of experiments.
Namely the first experiment was driving in a real car on expressways and streets, while music is played and mental conditions are measured including the behaviour of driving of the subjects. The advantage of course is that the data is gathered under real life situations, but the disadvantage is that for each test subject the situation could be different, due to possible different traffic conditions and weather conditions, which possibly could influence the data.
The second experiment also consisted of driving a car in real life while listening to music, but this time on a test track. With this kind of experiment, there is also the advantage that to some extent it reflects a real life situation, but because of the absence of other traffic, unexpected and undesirable events are taken out.
The third experiment consisted of driving a car in a virtual driving simulator on urban streets and highways while also listening to music. Also the driving behaviour and mental conditions are measured. The advantage of this simulator based study is that all unexpected and undesirable events are eliminated from the study, but on the other side this study doesn't reflect a real life situation.
The fourth experiment consisted of the study subjects driving a car in a video race car game, while also listening to music. The researchers suggest that the advantage of this method of testing is that the scores from the came could be used as data and that, as the other driving simulator, the unexpected and undesirable events are eliminated. The disadvantage of this method is that it hardly reflects any real life situations.
The researchers conclude that, while there is possibly selection bias because all test subjects are male University students in their 20's, the results show the following: Higher volume of music had a negative impact on driving; that music with a higher beat tempo also had a negative impact on the driving; while favourable music had good influence and unfavourable music had negative influence on driving. Further the researchers recommend that appropriate slow music with a low volume in a car should be played while driving, to reduce the potential negative impact of music in a car while driving. For further research studies the researchers suggested that a more diverse cohort of test subjects should be used to reduce selection bias and that the influence of music on driving should be tested under other driving conditions as well.

In a validations study [11] from 2015 an assessment was made to determine the scale of error in a driving simulator. A driver group of test subjects which were recruited, who were instructed to drive an on-road route in real-life and to drive the same route in a driving simulator. It's proposed that validating a driving simulator is best done with the real life on road situation. The study from 2015 specifically set out to validate the driver response in a simulator and if these responses were similar to the on-road situations. The study used a UC-win/Road program to assess the simulation, it was originally a software program which was used in urban planning, and traffic modelling among other uses. The driver sits in a car seat with a seat belt, steering

wheel pedals and instruments on the dashboard. The driver has 3 monitors in a 180 degrees angle in front of them, as well as for traffic sounds etc. a built-in audio system is used. Although the software appears to be well suited for civil planning and traffic simulation, there is ample literature available, describing the suitability of this software as a driving simulator for assessing driving performance. The following outcomes for the drivers both on the road and in the simulator were assessed: speed, vehicle control, road laws and road signs among others. When the two tests were compared in relation to driving errors, most assessment points didn't differ significantly from the on-road and driving simulator assessment. However the points that differ were maintaining the correct speed of the vehicle, staying in the correct driving lane, indicating to turn and the selection of the gap between the car and other objects such as curbs and other objects.
As mentioned before, because the number of points assessed significantly differ in both driving experiences, the driving assessment in the driving simulator doesn't well replicate the driving simulation in a real car on a road. Further the researchers suggest that to better understand this correlation, it is advisable to do similar studies as the one discussed here but with different driving scenarios. The researchers further conclude that for a number assessment points a simulator can be used, while also a larger sample size should be considered, as well as the inclusion of specific test subjects, such as drivers with visual impairment and older drivers.

The latest study [13] in this research review is from 2019. This study set out to answer the question if a driving simulator would help the driver to understand the road infrastructure better. Apart from describing different kind of simulation technologies, which were discussed in other research papers and,which were reviewed for this thesis research; it also sums up the advantages of driving simulators, which are: first the ability to perform tests in situations, which otherwise in real-life would be hazardous for the test subjects; second there is more control over the situation as unpredictable and random events can be ruled out; third the road and infrastructure changes can be easily implemented in the software; and fourthly collecting data at a high frequency will be easier from sensors on the drivers testing vehicle and in the 3D environment through coding to record for instance traffic data, and detection of actions of objects within the 3D environment, in addition to vehicle movements.

2.1 Object Oriented programming of 3D models

To better understand the process of Object Oriented programming, the discussion around the following illustration might help. Looking at for instance an arcade road racing system from the 80's, there can be seen for instance that the trees next to the road all look the same, such as illustrated in figure 2.1.

Figure 2.1, 1980's road racing game with identical tree models, Benjamin Shahrabani, 2015
https://petrolicious.com/articles/the-growing-nostalgia-for-classic-racing-games

Even though the code for the tree model would be relatively easy in this case, however this code would need to be re-entered for each tree in the main code. If many of the same objects are present in the software program, like the tree for the example above, it would make the code unnecessary long. Also for non object oriented languages, it is much harder to program different objects and at the same having the ability to interact with each other, as the objects written in non object oriented programming are all interwoven in the whole code and contain no code in terms of their dynamics or other parameters. Non object oriented models mostly contain only code on how the objects should be presented on the screen.
If the same example as presented in figure 25 would be taken, but instead of programming each tree with an object oriented language. It can be concluded that to program one tree there is more code necessary than for one tree in the non object oriented language. But the advantages of a tree programmed in an object oriented language is it can be reused many times over, without fundamentally making the code of the whole program longer. Programming in an Object Oriented language has another advantage, and that is that there is the ability to build in parameters in the Object Oriented model, and that by changing these parameters the size, structure, shape and colour ect. can be changed. So, going back to the tree example, if a number of parameters in the Object Oriented tree are built, there is the possibility to change the size of the tree, number of branches, leaves etc. simply by changing the value of the parameters of that tree model. To show how this works please look at figure 2.2 and 2.3. In these figures it can be seen that from an online available 3D tree model, a number of parameters of this tree can be changed by simply moving along the scale of these parameters.

In figure 2.2 can be seen that the tree as it is presented when opened in the website, and in figure 2.3 can be seen that after a number of parameters have been changed.

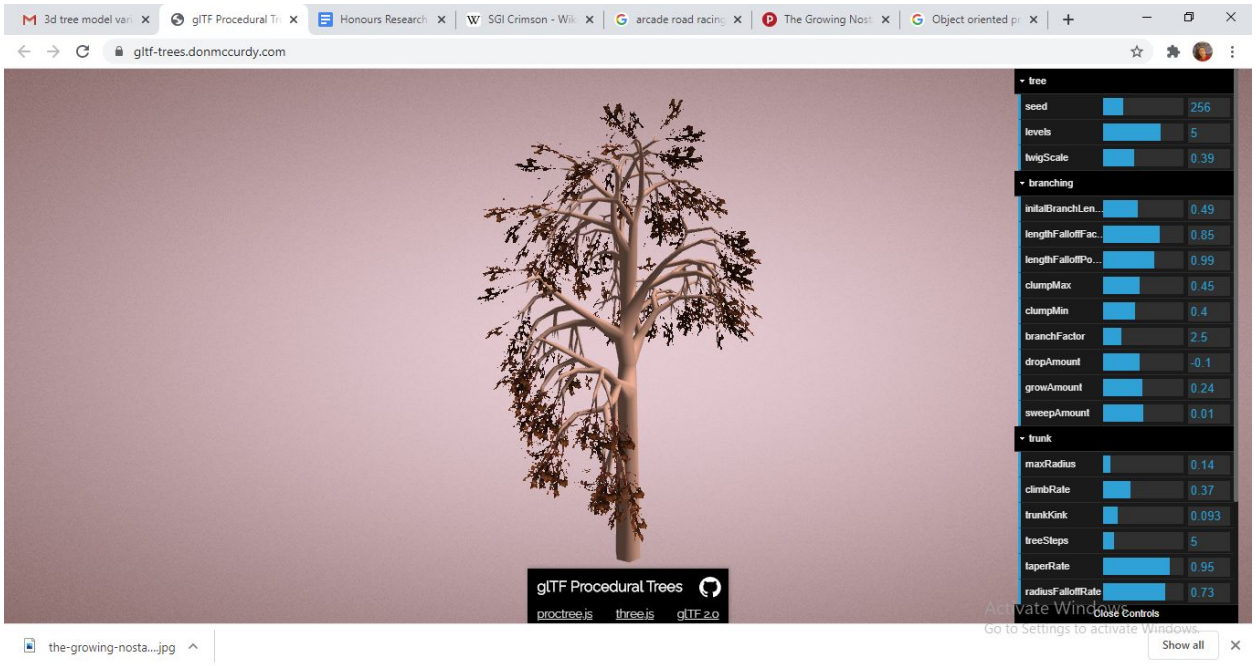

Figure 2.2, with unchanged parameters of which can be seen on the right side of the tree model, (Don McCurdy, GitHub, glTF Procedural Trees, 2020), https://gltf-trees.donmccurdy.com/

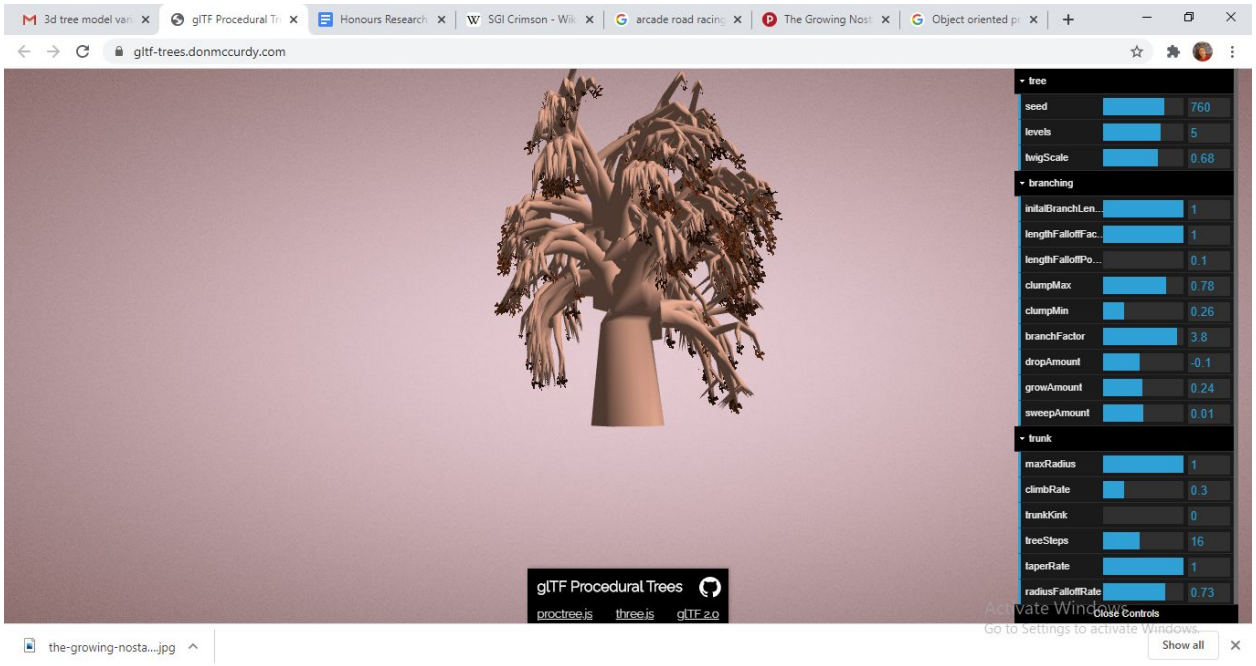

Figure 2.3, This is the same tree model as in figure 26, but now with a number of parameters adjusted. (Don McCurdy, GitHub, glTF Procedural Trees, 2020), https://gltf-trees.donmccurdy.com/

3 The 3D modelling process

A research project [3] from the Road Safety Commission in Western Australia (RSC) shows that from all the road fatalities in 2002 in Western Australia 3% of these fatalities consisted of bicyclists. This research project also indicated that in the 5 years leading up to 2002, fatalities of all road users decreased for the 5 year average, however the fatalities of bicyclists increased from an average of 5 fatalities to 6 fatalities for the year 2002.

This study also investigated the number of hospitalisations per road user group. The number of bicyclist hospitalisations amounted to 4% of the total cohort of road users, which were hospitalized. However the bicycle hospitalizations increased with 25% to a total of 121 hospitalizations in 2002 compared to the previous 5 year average hospitalization numbers. More recent and slightly different data [4] shows that in 2015 there were 3 bicycle fatalities in Western Australia, which is a decrease compared to the 5 year average. While the number of fatalities appear to be low, every death is one too many. Also with increasing traffic the fatalities could increase. With relatively cheap and easy interventions increased road safety and reduced traffic fatalities and injuries could be achieved.

The School Population and Global Health at The University of Western Australia in cooperation with Curtin University in Western Australia established a joint research project to better understand the road safety issues of bicyclists in Western Australia. The research approach of this joint venture is to investigate impact on bicycle behaviour with different road signs and indicators (different road treatments). By gathering data in relation to the response of the bicyclists to these different road treatments in a bicycle simulator, a potential better understanding of the impact on road safety of these situations for the bicyclists can be achieved.

The bicycle simulator consists of a bicycle mounted on a frame with 3 sensors attached to it. The sensors are: The first sensor is a speed sensor, which is used for registering the simulated speed of the test subject on the bicycle. The second sensor is a sensor which records at which angle the handlebar is positioned and finally the third sensor is a sensor which registers when the hand brake is applied. A very important part of the bicycle simulator are the 3D goggles, which are connected to the computer. The 3D goggles are worn by each test subject to create an realistic awareness of the 3D environment for the test subject. This realistic awareness is possible because the test subject can look around with the 3D goggles and while doing that the environment will change accordingly, thanks to two sensors connected to the computer, which sense the directional movement of the goggles.

The three sensors on the bicycle are connected to a Programmable Logic Controller (PLC) system, which processes the signals from the sensors and sends them to a computer. The computer runs a Virtual Reality (VR) software Package called Unity3D, which translates the signals from the PLC box to corresponding movements and surrounding changes in the virtual environment. The computer in turn sends video data to the 3D goggles, the view in the 3D

goggles changes to give the test subjects a better experience of the Virtual Reality 3D environment.

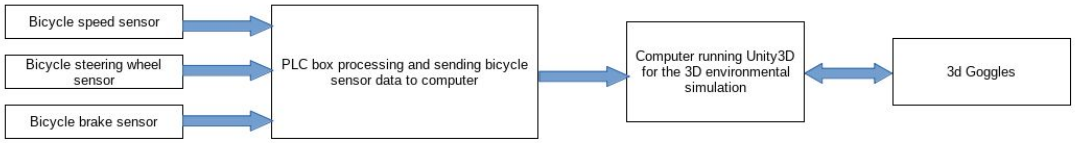

Figure 3.1, Blockdiagram of the bicycle simulator, (Machiel van der Stelt, 2020).

In figure 3.1, can be seen that the data transfer from the bicycle to the computer via the PLC box is one directional. This means that physical actions (speed, handlebar position and brake intensity for this bicycle simulator) on the bicycle are translated to digital signals which are processed by the computer to show a corresponding 3D virtual environment in the goggles. Ths one directional data design also means that there is no feedback from the computer to the bicycle. This means that there is no physical feedback to the bicycle, this means for instance that by riding on the bicycle simulator and in the case of a virtual crash against another object, or riding on a cobble stone surface; the physical awareness will be not experienced. However, the 3D goggles in combination with the computer have a feedback capability built in. As mentioned above, the 3D goggles, with the help of two sensors, are able to sense head movement, which the computer processes, resulting in 3D changes of the 3D environment in the googles. This capability is represented by a bi-rectional arrow in figure 3.1 between the goggles and the computer.

In the figure 3.2 below the physical set up of the bicycle simulator can be seen. The speed sensor is installed on the left side of the bicycle on the back axis and the bicycle frame. On the left side of the back axis there are alternating reflecting stickers installed, while on the frame of the bicycle a light sensor is installed. With every rotation of the bicycle wheel the parts of the axis with and without reflecting stickers passes the light sensor. With a timer and the data from the light sensor the speed of the bicycle can be calculated, which is done by the PLC box and the computer.

The brake system consists of a sensor, which is directly connected to the normal bicycle brake system. The use of the sensor in combination with the brake doesn’t interfere with using the brake in a normal manner, because the test subject doesn't feel the physical difference between a brake with or without a sensor. The brake sensor is designed in such a way that different grip or hand force applied by the test subject on the bicycle brake is translated into different signals. This enables the test subject to break hard or soft and have that corresponding action translated in the view of the 3D goggles.

The steering wheel sensor or rotary sensor is directly connected to the bottom of the fork of the bicycle frame and any steering wheel movements to any direction are directly registered by the rotary sensor, after processing by the PLC box the signal is then sent to the computer.

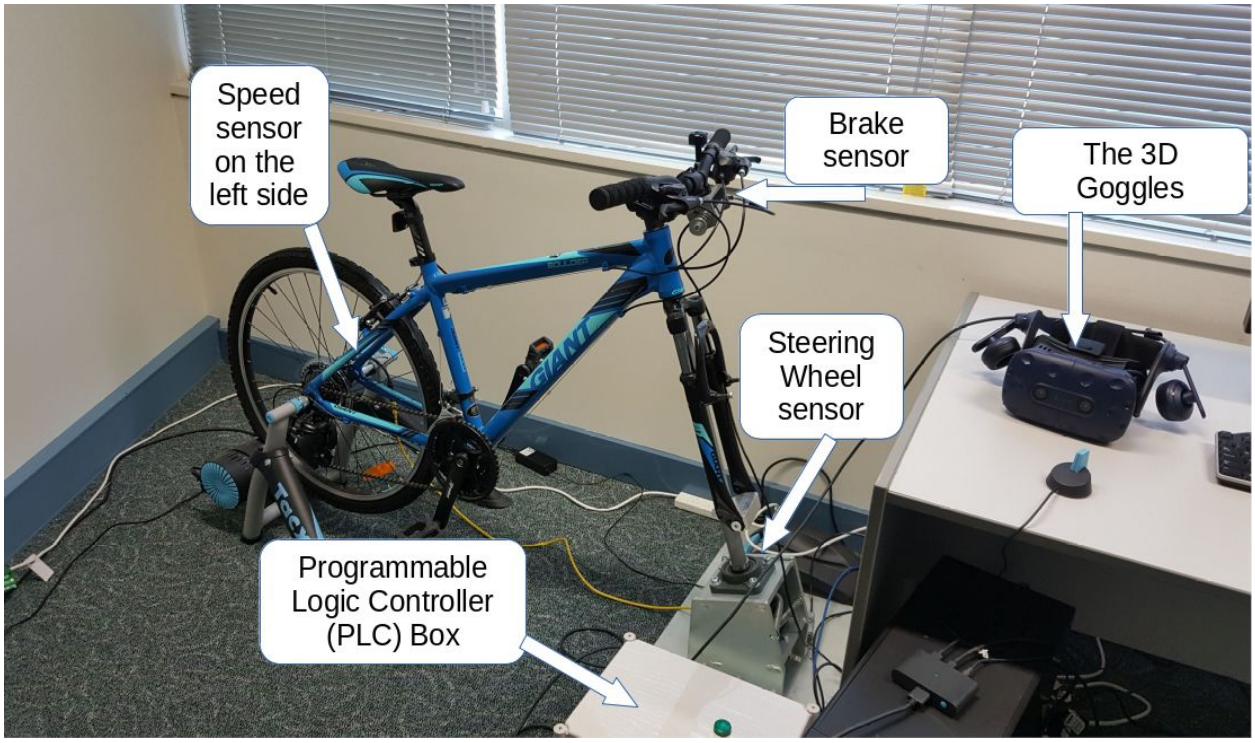

Figure 3.2, The bicycle simulator with the sensors, PLC box and 3D Goggles, (Machiel van der Stelt, 2020).

In figure 3.3 below the Unity3D Editor software package screenshot can be seen, which is the view when actively editing and testing 3D environments. In the screenshot at the bottom the editing and controlling area is pictured, there, files are selected, such as surface structure files or object files, which can be put in the active project. The top two screens are preview screens and correspond to two cameras placed in the 3D environment. One camera is placed next to the road, which looks at the bicyclist creating a third person view and in that third person preview the second camera placed on the head of the bicyclist can be seen, which creates the second preview, which can be seen in the screenshot, for a first person view.

Unity is a widely used editing and design program for either 3D or 2D environments [8], which are used to create content for games, engineering applications among others. In addition to that Unity3D is a relatively easy to use program to create 3D game or scientific environments for 3D goggles. However if there is a need to create specific objects or shapes with detailed and specific characteristics, then it is better to use a 3D design software package for that, because Unity3D is specially designed for game environments. In this case it would be helped to have access to a 3D software modelling package which has a wide array of road and traffic objects such as road signs, road treatments, cars and different designs of roads. Luckily, that package has already been identified by my advisor Thomas Braunl and his students when working on

other research projects. That software package is VectorZero's RoadRunner, which will be briefly discussed later in this paper and how it could be used for the bicycle simulator research project.

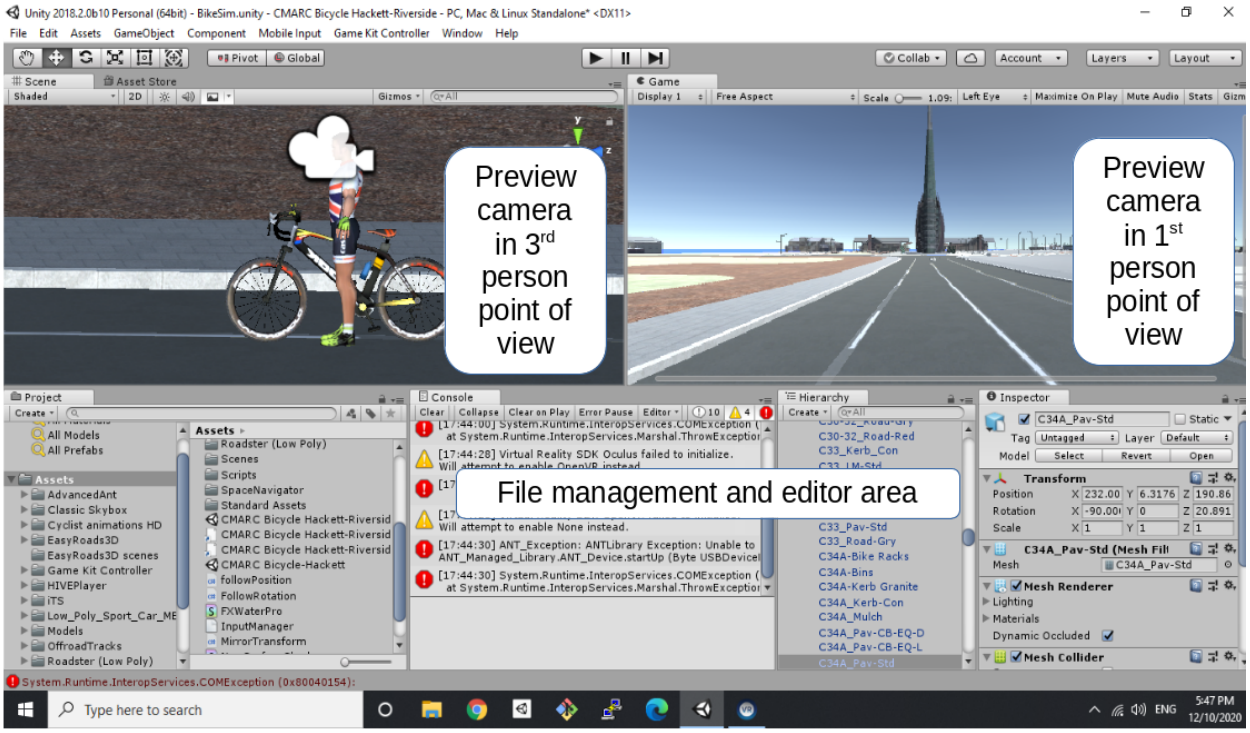

Figure 3.3, Unity3D Editor with two previews of the simulation engine, (Machiel van der Stelt, 2020).

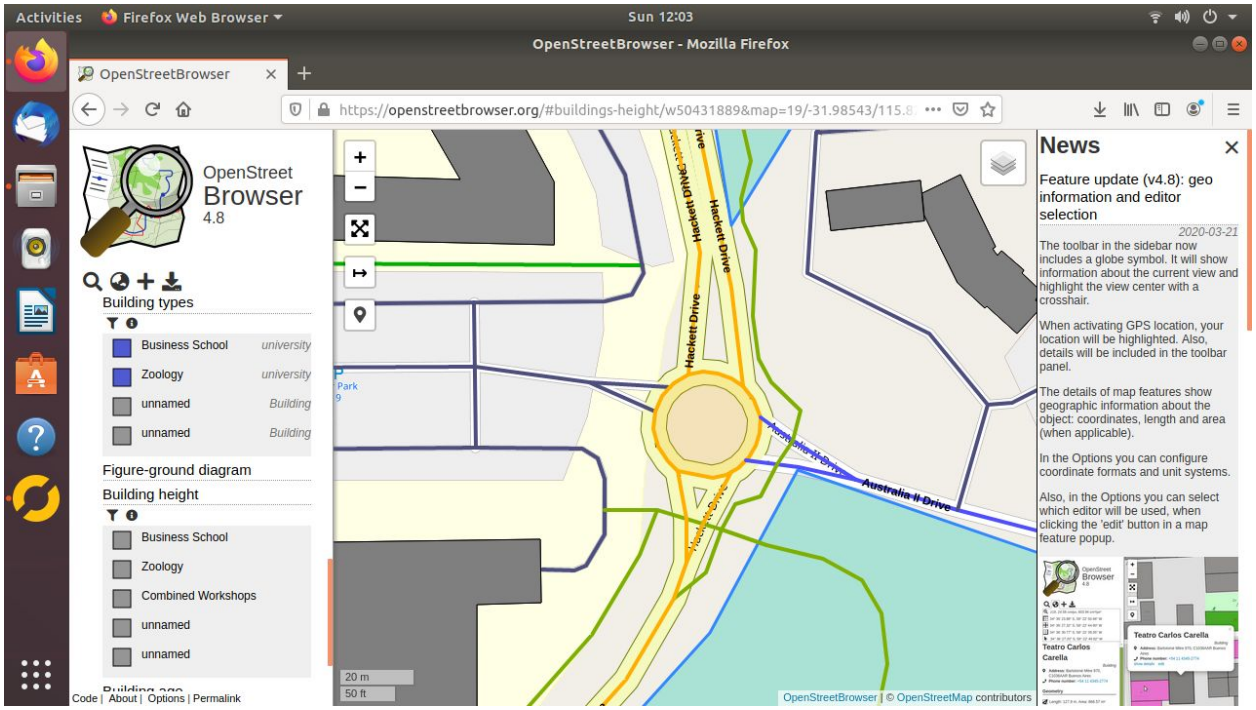

Figure 3.4, Hackett Drive located on OpenStreetMap on https://www.openstreetbrowser.org ,(screenshot, Machiel van der Stelt, 2020).

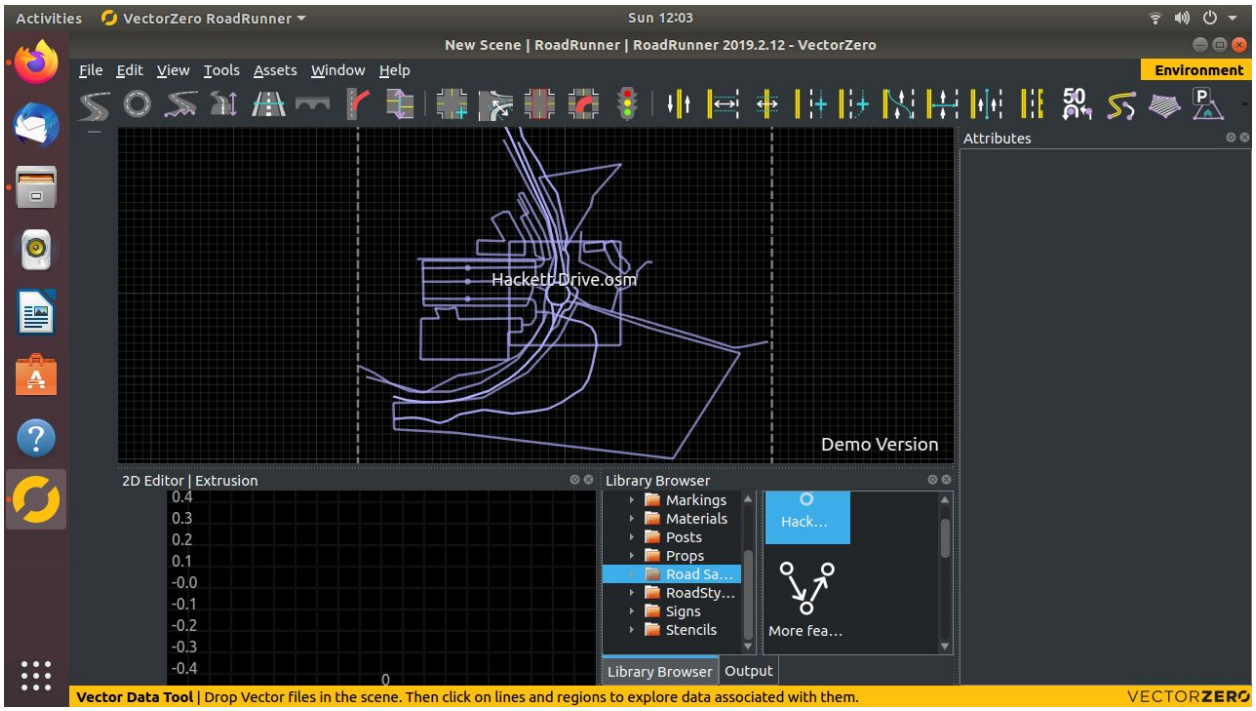

Figure 3.5, Hackett Drive OpenStreetMap data in RoadRunner in Top down view, screenshot, Machiel van der Stelt, 2020

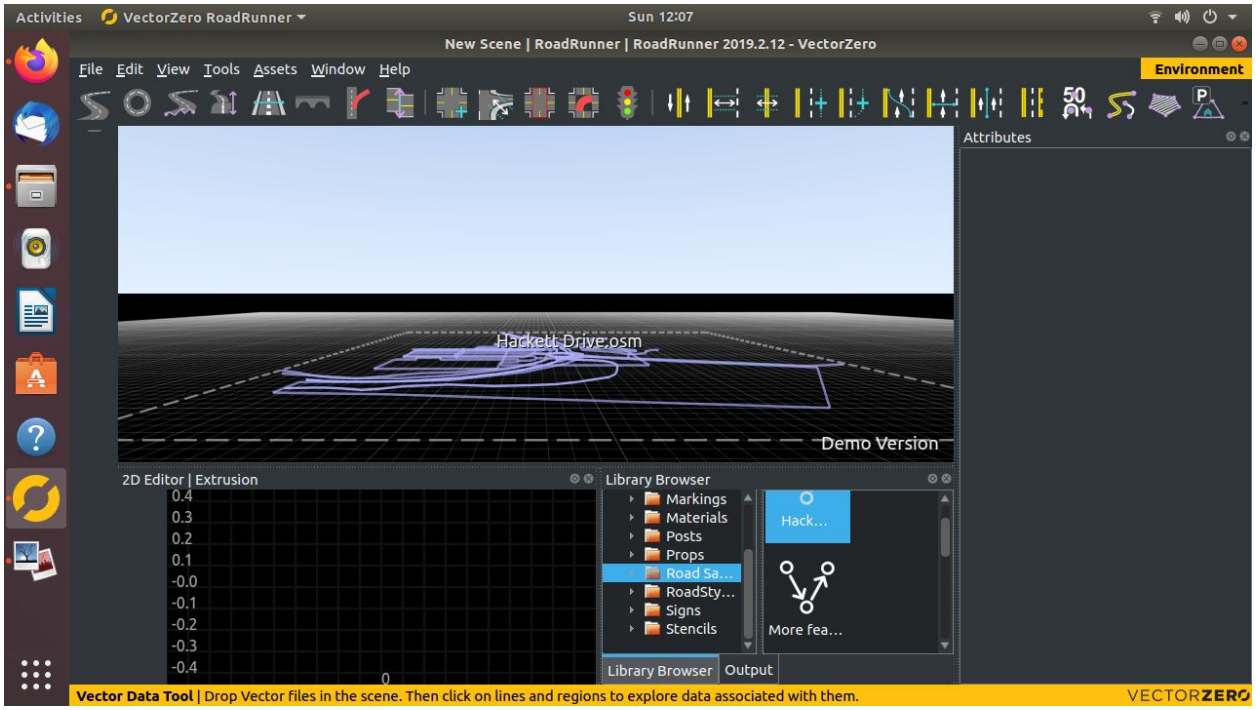

Figure 3.6, Hackett Drive OpenStreetMap data in RoadRunner in North view, (Machiel van der Stelt, 2020).

This thesis research is part of a Bicycle Simulator research cross-sectional study, which attempts to simulate the behaviour of cyclists in different road situations with different road treatments, such as signages on the pavement and different widths of bicycle paths. A number of characteristics of a Cross-Sectional study are: Some characteristics of a Cross-Sectional study are: multiple subjects are accessed individually, there is no control or comparison group, the study can be retrospective or prospective. The aim of this Road Safety study is to get a clearer picture if certain road treatments have an improvement on the outcome on the safety of cyclists.[5].

The scientists selected two locations to use in the study. The first location was a roundabout on Hackett Drive in Crawley, and Riverside Road in North Fremantle.
The roundabout on Hackett Drive was chosen so the bicyclists' behaviour can be measured when moving from the bicycle lane to the shared road just before the roundabout in both the treated and non treated road situations.
Riverside Road was chosen for its distinct long and wide curve with dedicated bicycle paths. For this section the aim was to see if the bicyclist stayed within the bicycle lane for both the treated and untreated situations. The untreated section had the current real life lane width of 1.2 metres, while the treated situation had a wider lane width of 1.4 metres.

A side visit was performed to get a clearer picture of the locale situations at both Hackett Drive and Riverside Road, specifications such as the exact width of the bicycle and car lanes, as well as curbs and signs at the side and in the middle of the road were analyzed.

For the bicycle safety research study 4 different road sections were created, the first one was Hackett Drive without road treatment, the second was Hackett Drive with road treatment, the third one was Riverside Road without wider bicycle lanes and the fourth one was Riverside Road with wider bicycle lanes. The four sections were called A, B, C and D. The researchers determined to create 4 different trajectories with each scenario having the different order of the A, B, C and D sections. The first trajectory had the A, B, C and D combination. The second trajectory had B, C, D and A scenario combinations, The third trajectory had C, D, A and B combinations while the fourth trajectory had the D, A, B and C combination. The four different trajectories with the different combinations of the locationsa, as selected by the researchers, are represented in Table 3.1.

Trajectory 01	A	B	C	D
Trajectory 02	B	C	D	A
Trajectory 03	C	D	A	B
Trajectory 04	D	A	B	C

Table 3.1: Four trajectories, with the four scenario combinations, (Michelle Fraser, Lynn Meuleners, 2020).

One weakness of prospective studies can be selection bias. Selection bias is that the selection of participants who are going to be involved in the study are selected in one or more conditions, instead of a random pick of participants. To help alleviate selection bias the researchers assign participants randomly to one of the scenarios.
After riding with the bicycle simulator over a short test track, to make the bicyclists get accustomed to using the bicycle simulator, the actual track will be entered. Each trajectory is about 1826 meters in length as compared to the real life situation.

During each person's assessment the coordinates and speed of bicyclist will be recorded for the whole route in hard data as well as graphically with a track line behind the bicyclist in the 3D model. This data forms the basis for the researchers, who then are going to assess the outcome of each bicyclist after exposure to different road treatments and traffic conditions.
The first step in the method was analyzing the real-life situations and how it best could be translated to a 3D model.
After that a selection had to be made between two broadly available approaches; First approach is, developing the 3D model from scratch, and the second approach is, buy a ready built model.

Because of specific 3D requirements, such as special road treatments and traffic conditions, building the 3D models from scratch is the better option so that the 3D models can be easier customized to the research project requirements.

The second step is setting the performance criteria for the 3D model software. Three criteria were identified.

1. Software should be specifically designed for developing road infrastructure 3D models, so easy and quick implementation of road infrastructure can be realized.
2. The possibility of implementing traffic in the 3D model, so more realistic and complex traffic situations can be realized.
3. Files should be exportable in many different formats, so that the 3D models can be easily used in other software packages.

The third step is researching literature to find the best available software packages needed for translating the real-life situations into 3D situations and fulfilling the criteria as mentioned above and to understand 3D modelling better.

In creating the 3D models, site visits were performed as mentioned above, but also road information from OpenStreetMap and Google Street were used to build the 3D models as close as possible to the real situation, in terms of road layout, road angles, curves, and road signs. The OpenStreetMap data existed of lines to indicate where roads, grass, build-up and waterways are located. In figure 3.4, 3.5 and 3.6 the graphical road data of Hackett Drive. Geo Spatial data was not taken into account because the height differences were small, also bicycling uphill or downhill couldn't be physically simulated on the bicycle simulator and could have introduced confounding in the research study.

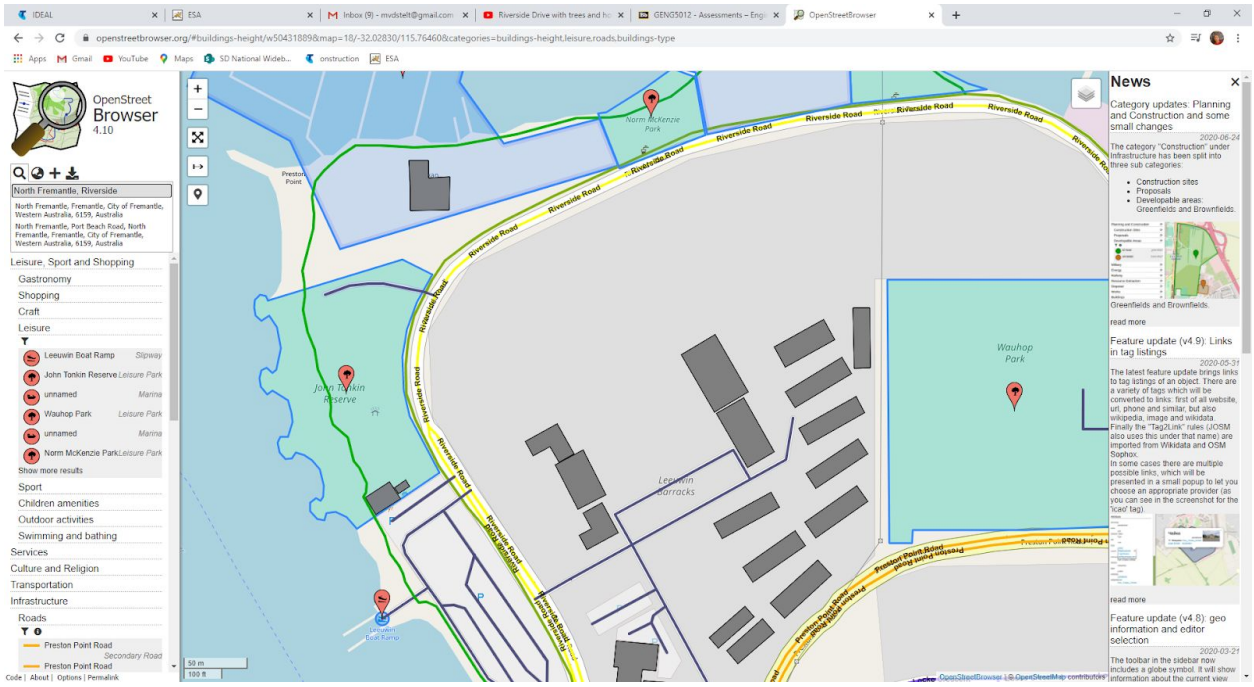

Figure 3.7, Top view of Riverside Road in OpenStreet map Browser, (Machiel van der Stelt, 2020).

The top view of the Riverside Road section from OpenStreet Maps can be seen in figure 3.7, which as mentioned earlier was selected by the researcher for use in the Bicycle safety research project. The colours of the lines, which can be observed on the map, indicate the different features and objects on the map, and also show the contours and borders of and between the different features. For instance the yellow lines show the location and shape of the roads, while the blue lines show the green or park areas.

In the upper left corner just under the magnifying glass of the Open Street Browser of figure 3.7 can be seen the ability to download the Map data for free. The data can be downloaded in three different formats, namely GeoJSON, OSM XML and OSM JSON. Different formats were tried out, and the findings were that the JSON format had more features included, such as surface structure and other details in the data, which made it easier to create the 3D models.

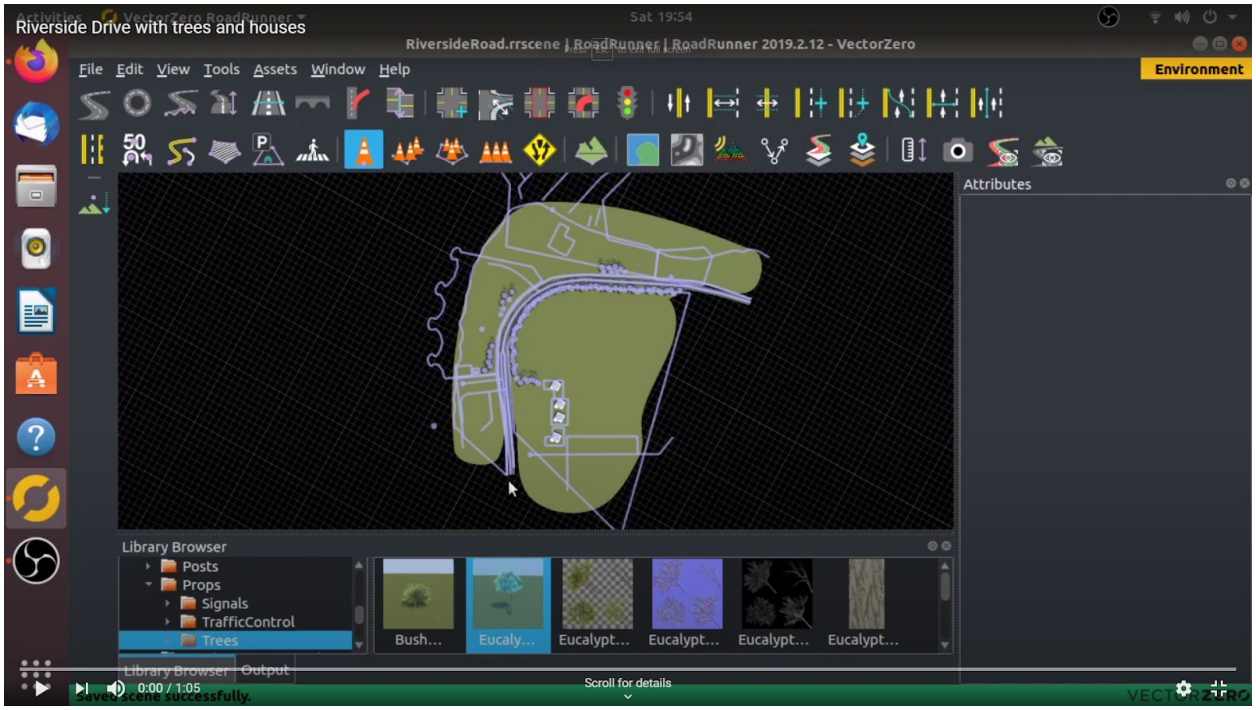

Figure 3.8, Topview Riverside Road with OpenStreetMap data and begin of building the 3D model, (Machiel van der Stelt, 2020).

The OpenstreetMap data was imported into Vector/ Zero's Roadrunner and the early steps of building the 3D model with the road infrastructure and grass areas, as can be seen in figure 3.8. Also the purple lines for the contours of areas and locations and shapes of the road can be seen. Because the import of the OpenStreetMap all the lines of the data turn purple, it's important when building the 3D model that the OpenStreetmap in the browser as represented in figure 3.7, is used as a reference. This becomes even more important when the data from OpenStreetMap is complex, in other words when there are many purple lines in close proximity.

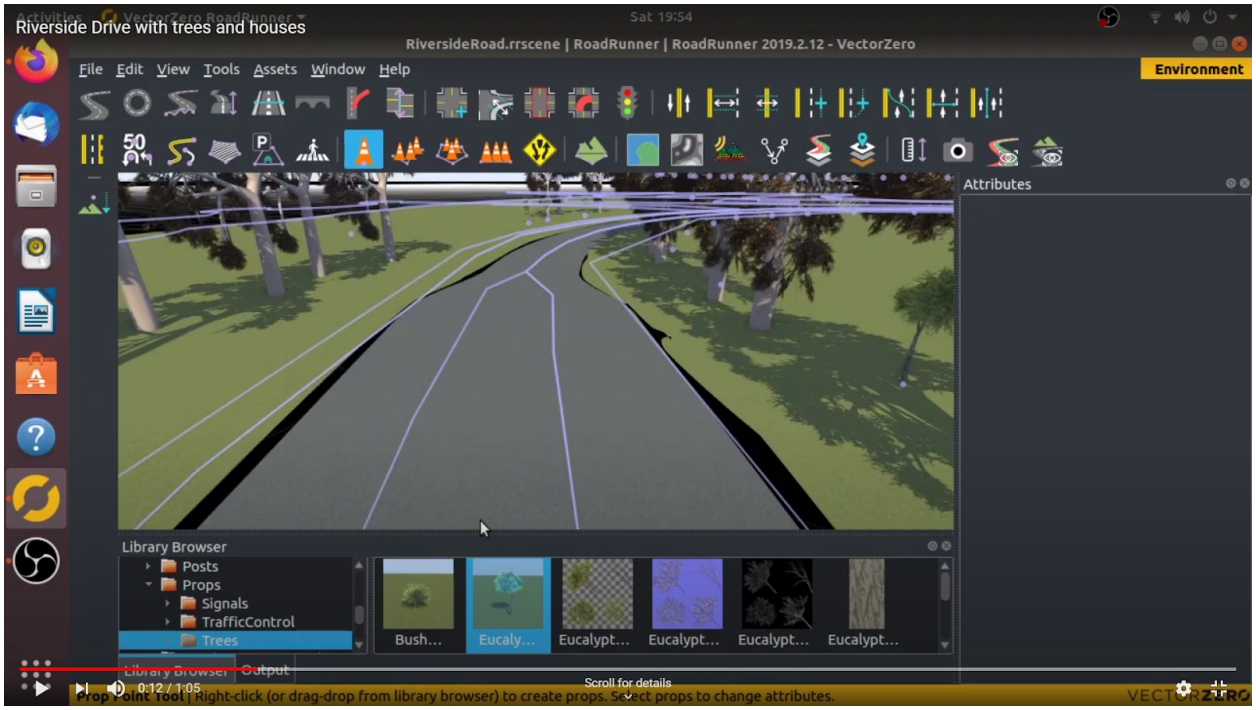

Figure 3.9, First step with the road applied and trees, (Machiel van der Stelt, 2020).

In Figure 3.9 the first steps are shown of applying road infrastructure on the basic OpenStreet map, which was already represented in figures 3.5 and 3.6. The purple lines in the middle in figure 3.9 represent the centre where the road and the purple lines right and left from it represent the borders of the road. Also clearly can be seen that the road narrows where the two purple lines merge.
Left and right from the road border lines the grass sections with trees can be observed. At this stage the curbs, road lines, bicycle lanes and buildings are not implemented yet. It is important at this stage to get the layout of the road as close as possible to the real life situation, as represented by the purple lines.

After the main layout has been put in the 3D model, such as the roads and the grass areas around the roads. More details can be added, such as bicycle lanes, curbs, road lines, road signage on the road surface and on the poles for roadside signs. For this stage it is useful to use either or both Google street and site visits to get a detailed impression of the local situation and details.
In figure 3.10 is pictured that the first bicycle lanes are added on both sides of the road, with a white road line separating both the road and the bicycle lane. While RoadRunner doesn't have the function of adding a bicycle lane, it was possible to add a footpath and give that the surface of the road with a bicycle lane instead.The Open Source photo editing software package GIMP was used to change the colour balance in the grey road surface picture. Changing the colour was done till it best matched the colour of a typical bicycle lane. After that the picture was imported into RoadRunner and applied to the grey road surface of the foodpath, to get the red bicycle path as a result. The height of the foodpath can be adjusted when it is already added to the road infrastructure, the height was left the same as the road. Adding the road line between

the foodpath and the road was an easy task. The area between the bicycle path and the road could be easily used to put on the road lane, and was not solely reserved for curbs only.

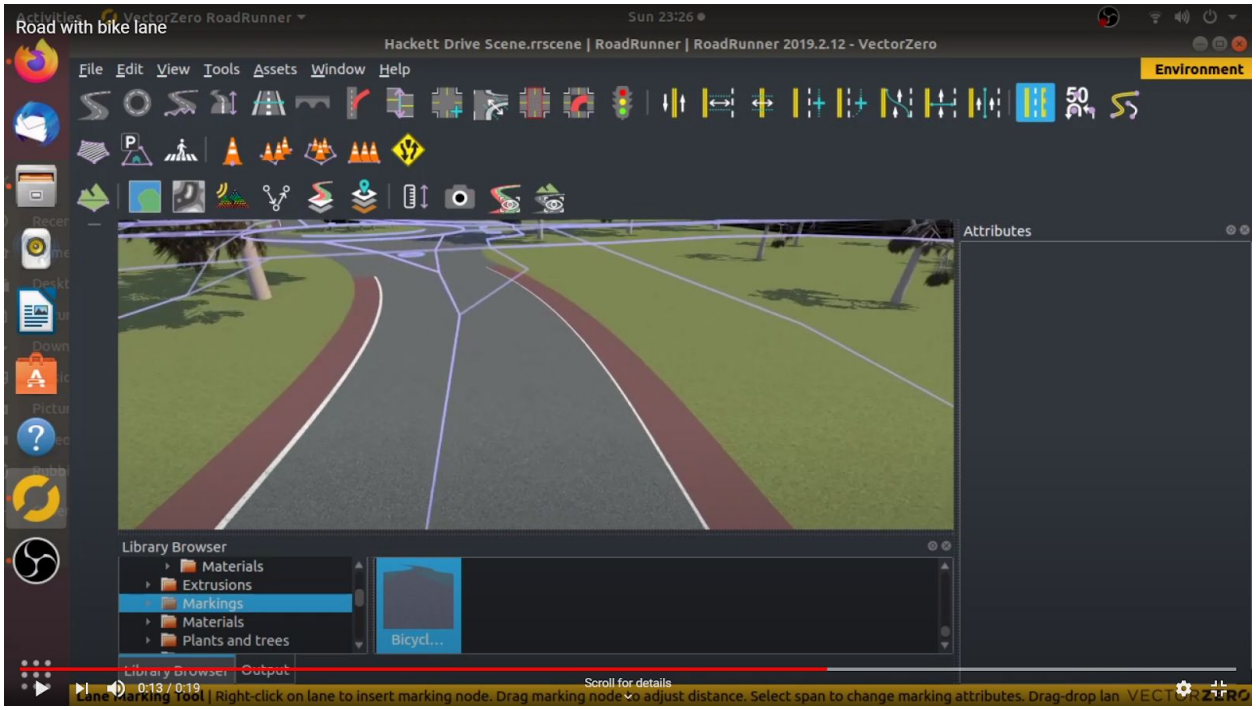

Figure 3.10, Roads have now basic bicycle lanes added, (Machiel van der Stelt, 2020).

As can be seen in Figure 3.10 the bicycle lane abruptly stops. In real life that of course looks different. Mostly the bicycle lane gradually narrows and then eventually merge with the road where needed.
Just like the bicycle lane itself, this gradual narrowing was exactly built in RoadRunner according to the real life situation and research project requirements. What was done was that the road where the bicycle lane ends was cut, then another short footpath was added to it and this short footpath gradually narrowed towards the curb. Because the colour and structure are the same, it appears to be the same lane, but in reality they are separate. The same solution was applied on the other side of the road and the other side of the roundabout. This approach was used because otherwise the bicycle lane would narrow towards the curb over the full length of the bicycle path, but the requirement was that the narrowing should only happen in the last few metres.

Further online research into finding Photogrammetry software for on the PC, the attention fell on the application called Meshroom. This is an open source free software application, which was developed by a number of university research institutes in Europe.
The requirements for the software to work well are: that a relatively powerful computer needs to be used, but more important the computer needs to have a Graphical Processing Unit (GPU) produced by Nvidia in order to be able to work.

Results from Meshroom, after getting used to taking pictures around an object, were surprisingly of good quality, both in model details and dimensions compared to the real life object. Figure 3.13 shows a picture of the 3D model of a Road Island, which was created with 30+ smartphone pictures and after about 20+ minutes of rendering on the PC. This 3D model can be easily viewed from all directions, zoomed-in and zoomed-out and added to other existing 3D models, in other 3D modelling software applications.

Because the available roundabout roadside sign model, which was available in the library of Unity3D modelling software, the arrows were going the wrong direction, an attempt was made to scan a real-life roundabout pole sign. But this resulted undesirable results, since the pole of the sign is reflective and disappeared in the created 3D model and had an overflow error effect on the sign itself.
The idea to create 3D models of real life objects, an investigation of 3D scanning techniques was undertaken. There exist 3D laser scanners, which generally scans rotating objects on the scanner while lasers detect the contours of the object. These laser scanners can quite accurately scan objects, but often lack the textures/information like colours. The surface textures/colours is essential information, which is needed for the 3D models. This issue makes the laser scanners ineligible to use for this thesis research project. Apart from the just mentioned limitation, the 3D laser scanner can only scan objects about the size of the scanner itself if a bigger object needs to be scanned, then a few metres the laser scanner gets very expensive. Handheld laser scanners are expensive as well.
After online research then the attention fel on another 3D scanning technique, which is called photogrammetry. The technique works as follows. At least 20 pictures are taken from an object with a smartphone or digital camera, while walking in a circular movement around the object of interest. The photogrammetry software then stitches the pictures together into an 3D object which can be used and manipulated in 3D modelling software.

This software uses the changing field of view around the object and accelerometer (if this sensor is present) information from the camera. The changing field of view is used by the software to determine what background is and what the object of interest is. The accelerometer is used to measure the direction of the camera movement, while walking around the object when taking pictures, so the location of all the photos in respect to each other and the object of interest can be determined. The accelerometer and field of view change are the main sources of information used to create the 3D model.
There are a few dilemmas with this technique and that is that shiny or reflecting objects result in errors in the resulting 3D objects, which are then not usable. This dilemma stems from the phenomenon that reflecting and shining objects tend to reflect field of view changes from views outside of the picture and/or reflect field of view changes on the wrong location picturewise.

There exist a number of paid and free Apps on smartphones with the ability to take pictures of an object and the software creates a 3D model of the object of interest. A number of the most popular or highest rating Apps were investigated but turned out to be generating undesirable results, either due to the software not being sophisticated enough and/or that the hardware where the software was running on (in this case the smartphones) was not powerful enough.

This eventually resulted in Meshroom not being used for creating 3D models, but instead the decision was made to use a picture taken from the front of the roundabout roadside sign and build that to a roadside sign pole, because it was discovered that RoadRunner has a built in function, with the ability to change a picture into a 3D file format object. Meaning the picture is still 2D as a 3D model, but can now be used to add to other 3D models. RoadRunner does have sign poles in its 3D library. This created the opportunity for us to use these poles and add the 2D in 3D file format signs to the sign poles. This approach of creating roundabout sign poles had a very good outcome in creating the 3D models of the roundabout signs in terms of detail and suitability (the roundabout signs didn't stick out as odd objects because the picture quality was the same and the model had the same level of detail) in the 3D Hackett Drive model.

Figure 3.11, the Bicycle lane gradually narrows until it disappears from the road, (Machiel van der Stelt, 2020).

3.1 The photogrammetry technique for 3D modelling

For the purposes of background information, the process and technique of photogrammetry for the creation of 3D models will be briefly discussed.
As mentioned before, the technique in broad lines consists of the software determining the object of interest from 20 to 40 or more pictures. This is done through the change of field determination. The software has the capacity to separate the background from the object of interest. When pictures are taken around an object, the background constantly changes, while the object of interest, stays large in the view of the camera. With this information the software

extracts the object of interest from the picture, and stitches the pictures together and creates a 3D polygon Mesh model.

There are a number of 3D file formats, such as OBJ which is an Open Source format, STL and FBX and generally consist of a polygon mesh to represent the 3D object. These polygon mesh files do not contain any information about possible moving objects. These files only contain information in regards to the shape and the surface of objects. Generally the surface in a 3D model file consists of triangles with different sizes put together either straight or in an angle along their edges, depending on the shape of the 3D model. All the triangles put together will create the 3D surface rather than volumes. Polygon meshes are also capable of representing volumes, in this case Volumetric Meshes are used to represent internal structure of 3D models. Figure 3.12 shows an example of a polygon 3D mesh model of the Dolphin.

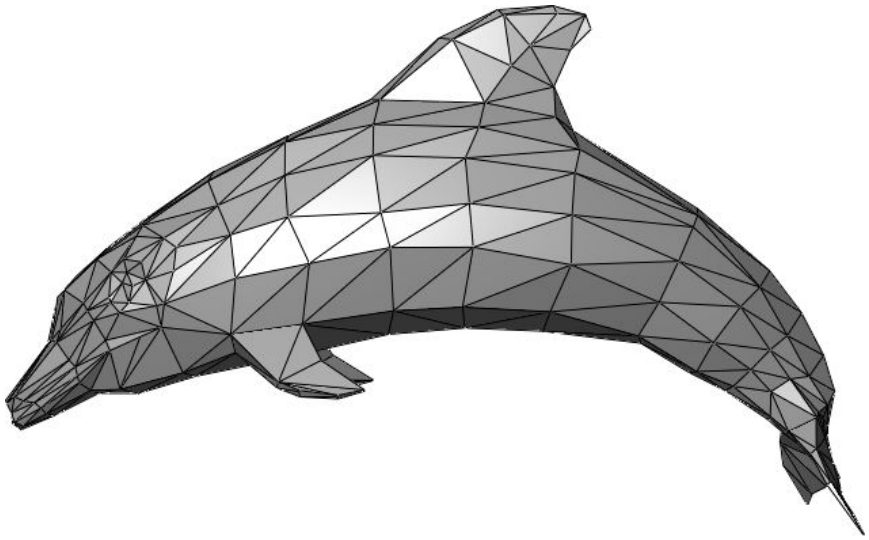

Figure 3.12.1 Polygon 3D Mesh model of a Dolphin, (Wikipedia, 2020).

The smaller the triangles of the 3D polygon mesh model are the higher the resolution of the 3D model.
As can be seen from figure 3.12.1 the triangles in one single model can have different sizes and shapes, that enables the software to generate asymmetric 3D polygon mesh models.
Generally 3D polygon mesh models consist of a number of files, working together. There is the main file which is the file with the actual information to build the 3D polygon mesh model. Additionally this main file also consists of mapping information, in regards to the surface or structure of the 3D model. This mapping is used to map the surface which is cut up in the same amount and sizes triangles as the 3D polygon mesh model, back onto the 3D polygon mesh model.

The main 3D polygon mesh model from the road Island and the related files are illustrated in figure 3.12.2. Figure 3.12.2 shows that there is one file with the obj extension, this is the 3D polygon mesh model file, which also has mapping information in regards to how the small

surface or structure photos are placed on each mesh of the 3D mesh model. It can be observed that the three files with the surface information have different sizes of the surface pictures. These sizes are related to the resolution of the 3D mesh model. The higher the resolution the more surface files there will be and of smaller sizes than of the surface pictures will be present in one of the files.

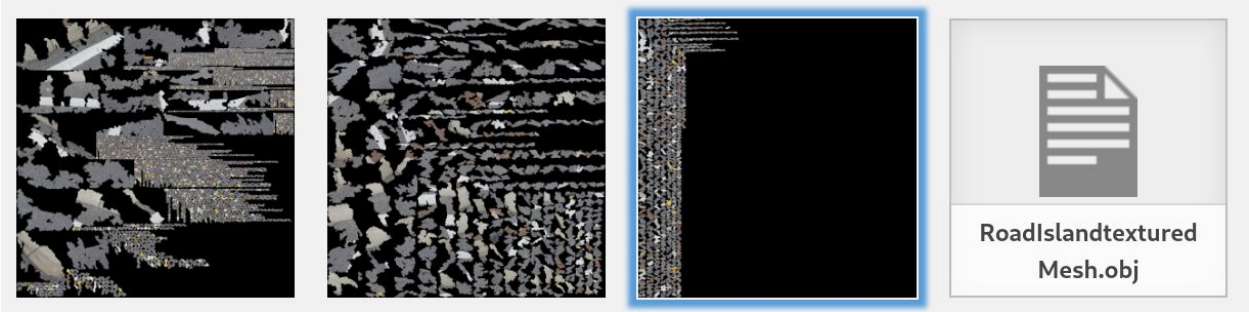

Figure 3.12.2 3D polygon mesh model file with three surface/structure files, (Machiel van der Stelt, 2020).

As mentioned before, Meshroom was used to try to recreate 3D objects which were not readily available in RoadRunner. For instance the recreation of a Road Island, was not able to be generated with RoadRunner. A visit to a local road island more than 30 pictures were taken from all directions with a smartphone and put in the Meshroom software application. In picture 13 can be seen that the result is very photo realistic. The model can be viewed from all directions, zoomed-in and out Also can be seen that the software accurately shows where each picture was taken, by showing the camera locations and positions around the 3D model.
However after the 3D model of the Road Island was imported into RoadRunner, it can be observed that the colours of the road island model disappeared. Also can be seen that after the road island 3D model was resized it looked misshaped. This misshape was due to the fact that the model had to up be scaled not equally in all three x,y, and z dimensions.
An additional issue was that thet x,y and z axis of the 3D road island model were not aligned with the x, y, z axis of the whole 3D environmental model. This becomes clear when having a very careful look and see parts of the Roundabout pole sign going in an angle sighways. These issues and the reason that the roundabout looks a bit of a different quality of the rest of the objects in the whole 3D environmental model, the decision was made not to use this model and the Meshroom modelling technique.
In real life there is a big road island on the spot where the scanned road island was planned in the 3D environment, the easiest way now to fill that gap was to put concrete and grass on that spot. Concrete with grass instead of an road island won't have an adverse effect on the quality of the data of the bicycle safety study, as bicyclist performance is not affected by a different coloured object in the middle of the road, while the bicyclist rides on the outside of the road on the bicycle path.

In figure 3.15 there is concrete placed where initially the scanned Road Island was intended pictured, but which ended up being a void after no object was placed. Therefore the decision

was made to put a concrete layer in the middle of the road. Also some grass was added to the road island to make it look similar to the rest of the environment. The result of putting grass on the road island on top of the concrete can be seen, the new road island appears to fit very well with the rest of the environment on the sides of the roads, as can be seen in figure 3.16. This same technique was applied to the adjacent road on the other side of the roundabout. RoadRunner has the ability to adjust the height of an object individually from other objects. Therefore the grass was stacked on top of the concrete, which in turn needed to be a slightly higher than the road in order to cover the road and the empty areas.

Figure 3.13, A very detailed Road Island after about 20 minutes of rendering in Meshroom with 30+ smartphone pictures, (Machiel van der Stelt, 2020).

Figure 3.14, Detailed Road Island Imported into RoadRunner, (Machiel van der Stelt, 2020).

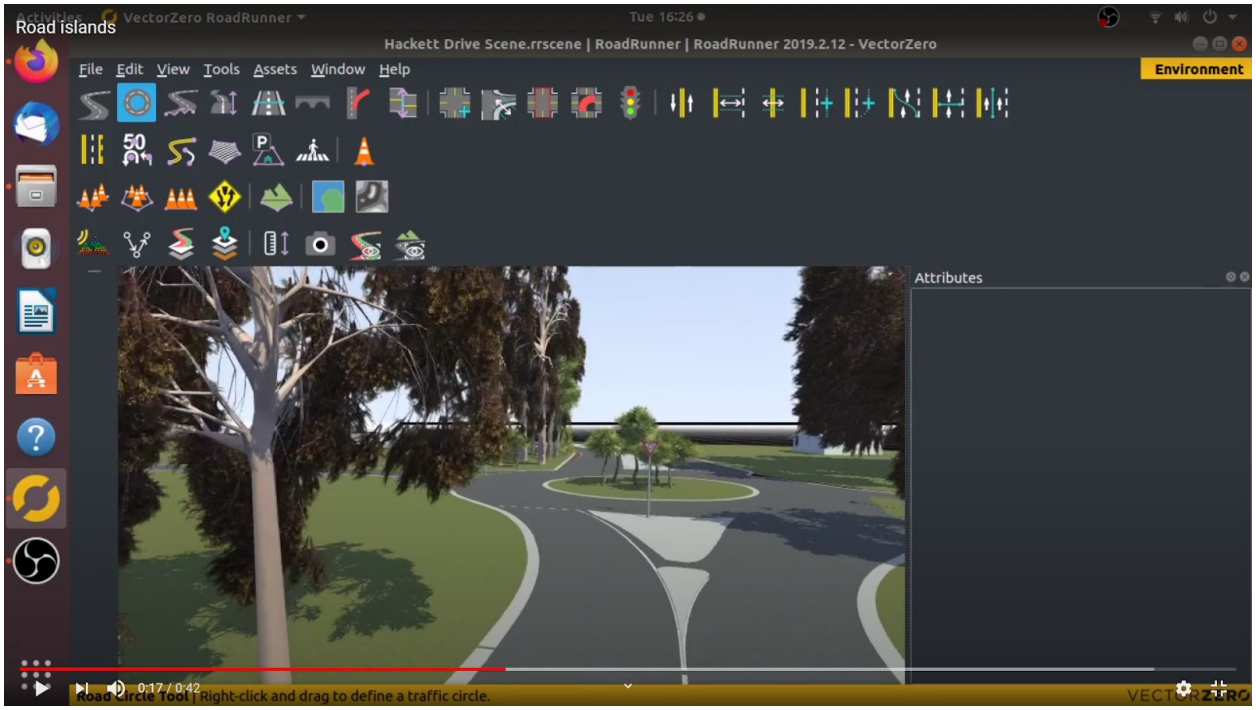

Figure 3.15, Road Island without grass, (Machiel van der Stelt, 2020).

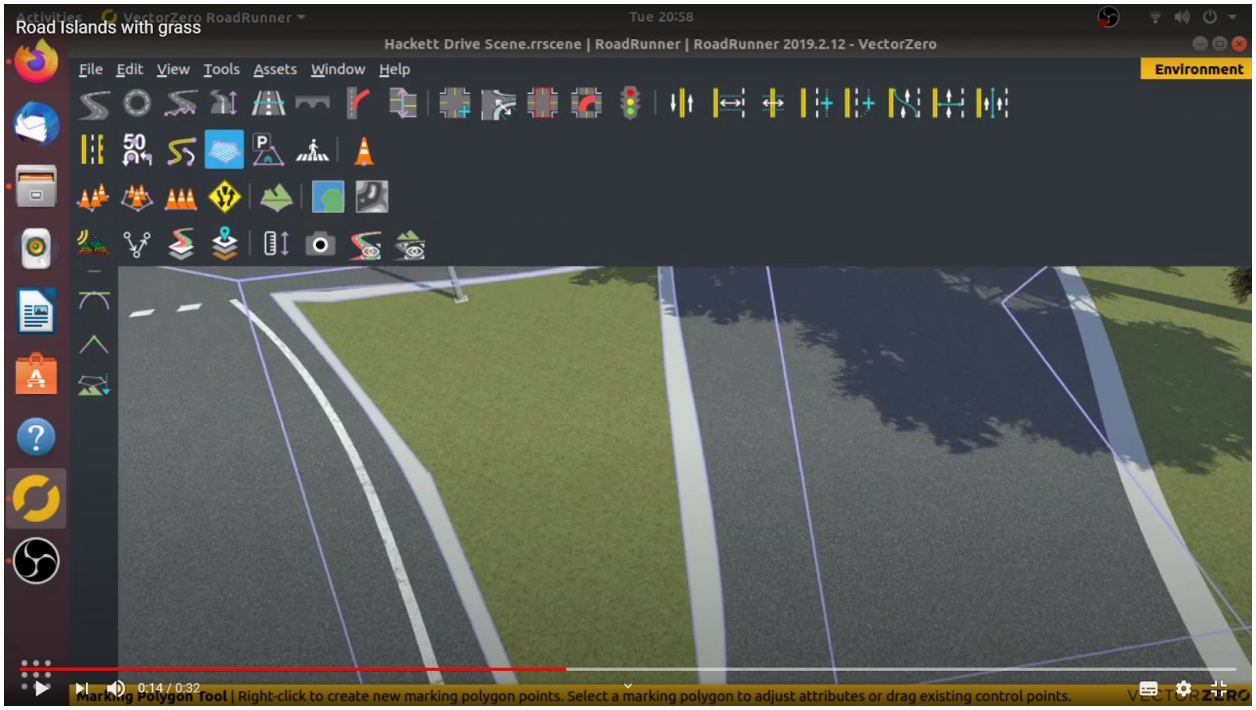

Figure 3.16, Road Island with grass, (Machiel van der Stelt, 2020).

As described before an attempt to import more realistic objects into RoadRunner of real life objects was made. Pictures were taken in a circle around the object and then processed by Meshroom. For the experimental approach in this research and out of interest also an attempt was made to produce a car 3D model with Meshroom. A life size car proved to be a bit too big to impractical to take pictures from around 30+ different angles, due to its size; also as mentioned before the shiny and reflecting surfaces prove to be a major issue to be able to reproduce a convincing 3D model.
To illustrate this issue with reflecting and shiny surfaces, an attempt was made to create a miniature 3D car with Meshroom. A toy car was put on the back porch, before about 20+ pictures were taken from the miniature car from different angles while the camera was moved in a circle around the car.
In figure 3.17 the 3D model of the miniature car can be clearly recognized, but the errors in the model are very obvious. The most obvious error for instance is that the surface of the 3D model car is not smooth, but shows bends and misshapes. Also, when the model from more directions is analyzed, different parts of the car merge, for instance in figure 3.17 it can be noticed that the bottom of the car is partially merged with the surface below the car, while in reality that area should be empty.

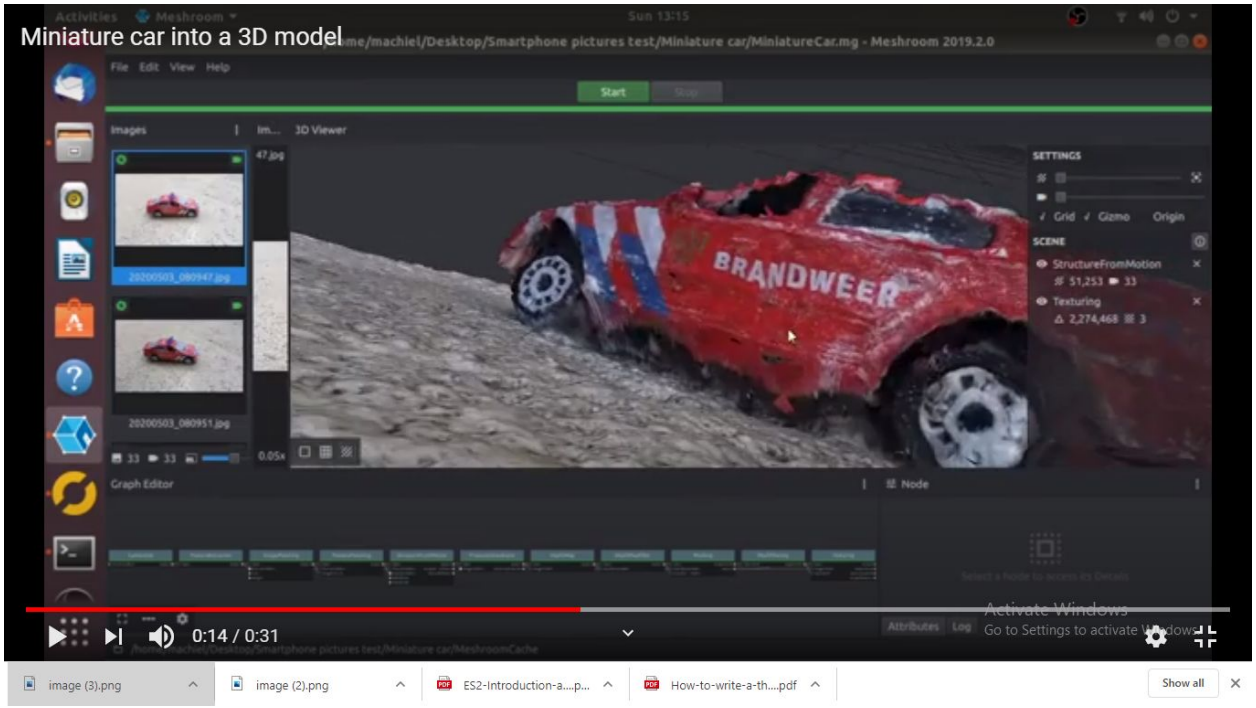

Figure 3.17, Miniature 3D car model produced after pictures were taken and processed in Meshroom, with obvious rendering errors, (Machiel van der Stelt, 2020).

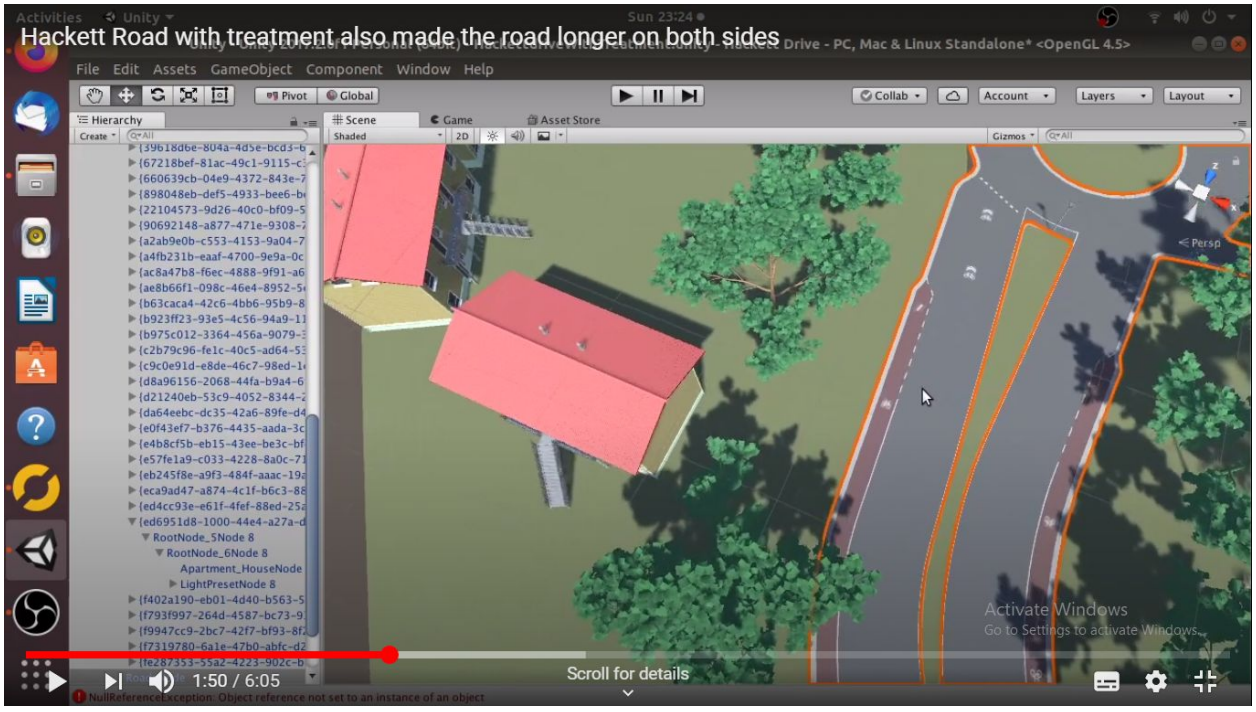

Figure 3.18, Road surface signs added to the 3D system, (Machiel van der Stelt, 2020).

In figure 3.18 a top view of Hackett Drive is shown, now from the other direction of the roundabout then in figures 14, 15 and 16. Also can be observed is that in the left bicycle lane a bicycle sign and an arrow have been added, which is used to indicate to the cyclists that they have to merge on the road and start using the same road as the cars. In the current real life situation the arrow is not there, the two sharrows closer to the roundabout in the real life situation are neither present. The sharrows indicate that on the road they are on, that the road should be used both by cars and cyclists.
The sharrows and the arrows are applied in the same way as they are used in an example in a real life situation on the East Coast as shown in figure 3.19.

When importing 3D objects from RoadRunner into Unity3D, there are occasions that the models show some errors. For example the tree, which was downloaded from the Internet, which was used without problems in RoadRunner, showed error, such as that the leaves were enclosed by a semi transparent rectangle. This error can be seen in figure 3.20, this was an unacceptable error and distraction. As there is no control over these external acquired 3D tree model designs. alternative 3D designs needed to be used. After researching the Internet, an available 3D model of a tree was found and proved to be suitable for use in both roadRunner and Unity3D. The model of this 3D tree can be seen in figure 3.18, with the top view looking down at the trees.

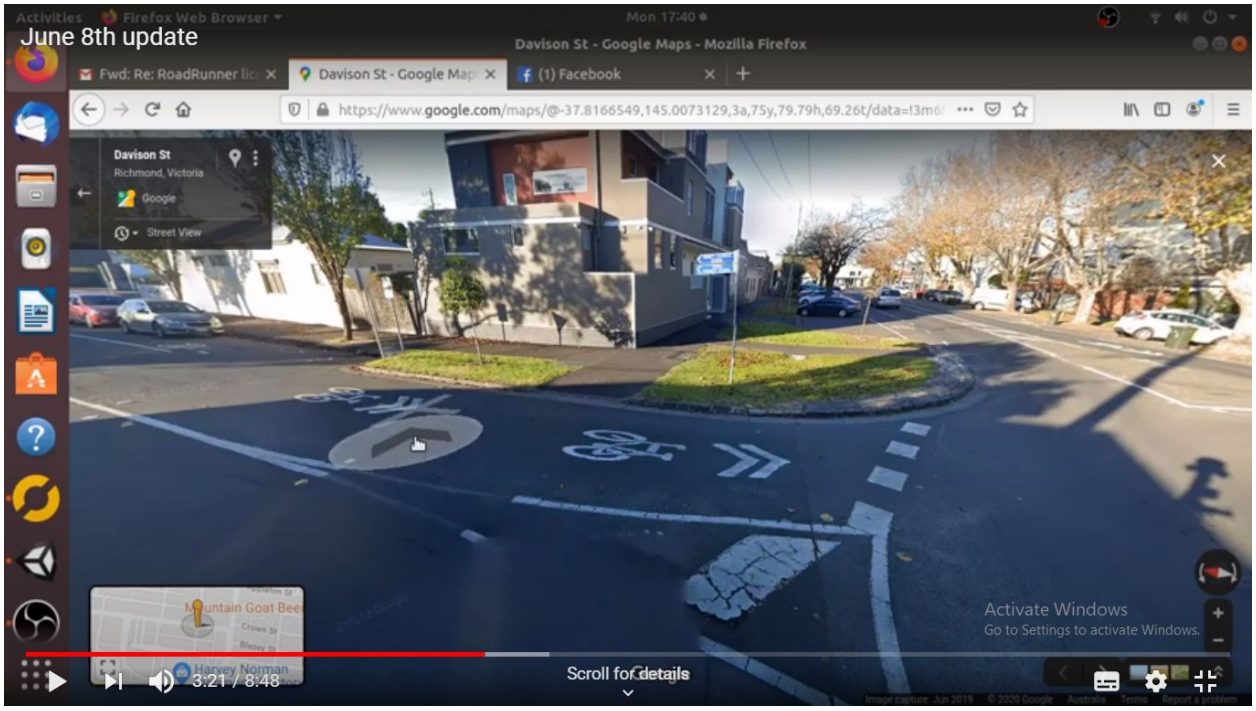

Figure 3.19, Real life situation used as an example for the modelling on Davidson St/Highett St in Richmond, Victoria, (Google Street, 2020).

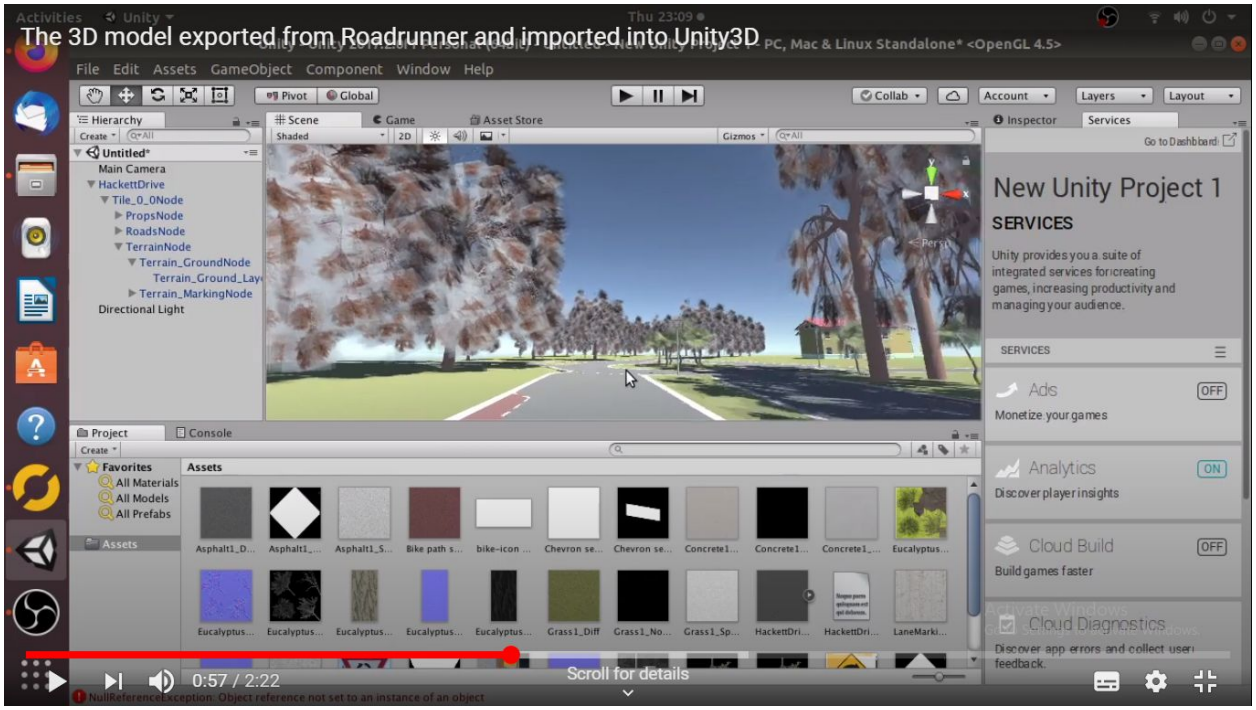

Figure 3.20, Trees imported into Unity3D with errors in the trees, (Machiel van der Stelt, 2020).

As discussed in relation to the Road Island issue when scanned with pictures, and the decision was made to use concrete and grass instead of the scanned island or the exact same design as in Real Life. An alternative design for Riverside Road was also used, where in Real Life over the length alternatingly Road Islands and road lines are applied. In the 3D model, in the middle of the road, the Road islands and Road lines were replaced by Chevrons, which had to be made from the ground up. The Chevrons are actually two road lines with reshaped rectangles, and which were each resized and were put in the same angle as used in real life. This approach, similar to the road island location, where the Road the bicycles don't ride in close proximity, should not affect the quality of the Bicycle safety study, and was for this reason approved by the scientists.

The Riverside Road and the Hackett Drive have different widths either because of the absence or presence of spacing between the two upcoming driving lanes and/or because of different widths between bicycle lanes. In figure 3.21 can be seen that the bicycle lanes of each road have different widths for which an adjustment had to be made. If looking carefully there can be observed that both roads at the meeting point are narrow and when moving away from that connection they both widen. This seems unnecessary, but this is an example to make sure all four sections can connect with each other in different orders, without the need of any editing to the road width.

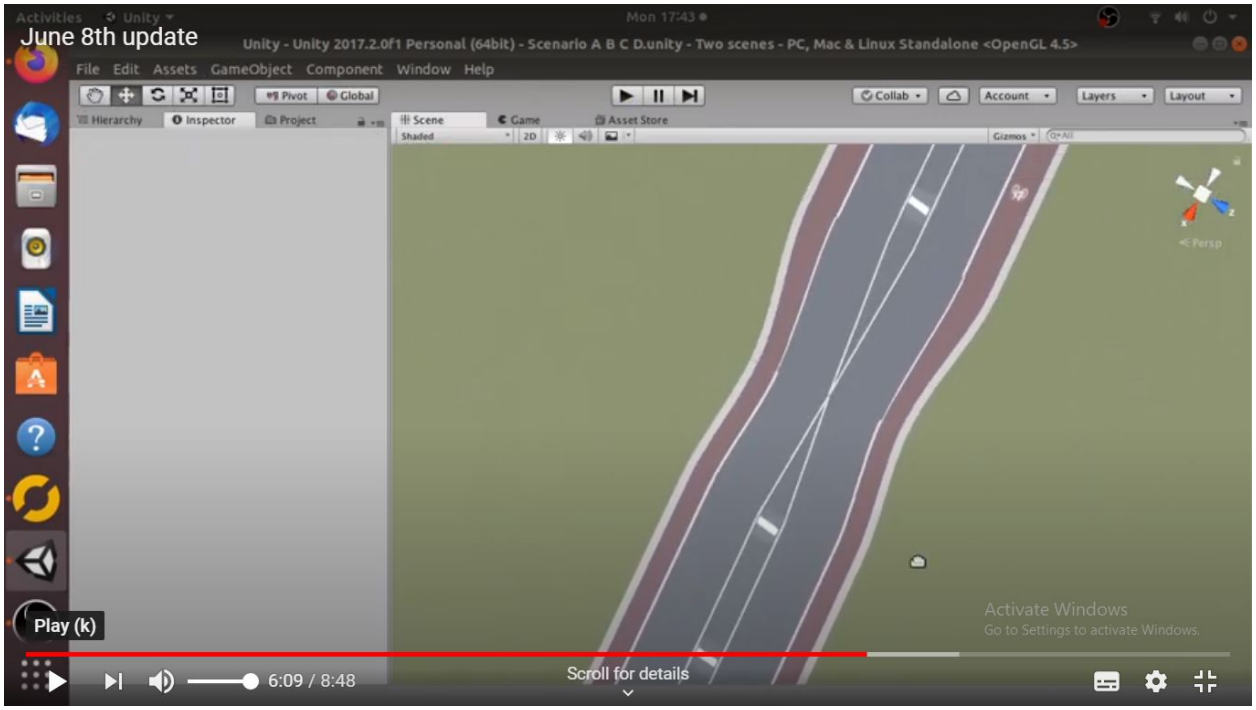

Figure 3.21, The end of each road section was adjusted to one standard width, (Machiel van der Stelt, 2020).

If zooming out of this section the example of the four sections connected in the A, B, C and D, as discussed earlier in this paper, order in Unity3D can be seen. The A section is for the Hackett Drive without treatment. The B section is for the Hackett Drive with treatment, for the C section Riverside Road without wider lanes is used and finally for the D section Riverside Road with wider bicycle lanes is used.

For making a comparison of the 3D virtual environment with the real life situation, a Google street screenshot of Hackett Drive was made, facing North and for the same location in RoadRunner.also facing North, alos a screenshot was created. Figure 3.22 shows the Google road for the real life situation, while Figure 3.23 shows the RoadRunner simulator result.

Figure 3.22, Google street screenshot of Hackett Drive going North to the roundabout, (Google Street, 2020)

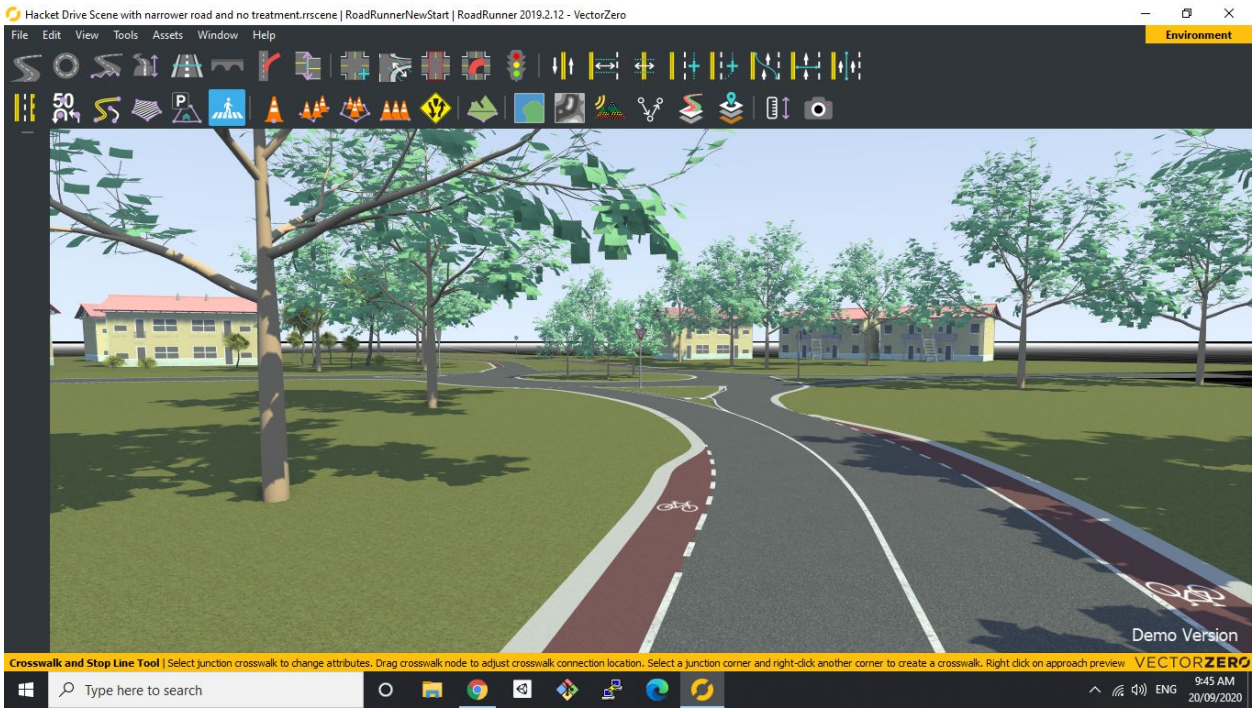

Figure 3.23, RoadRunner screenshot of Hackett Drive going North to the roundabout,)(Machiel van der Stelt, 2020).)

Comparing the roads in figures 3.22 and 3.23, it can be seen that the road and bicycle lanes have a close match in terms of width, length, proportions as well as the angles and curves of the roads. As discussed with the researchers, the surroundings in the 3D model didn't need to have a close match with the surroundings of the real life situation. This is mainly because there will be only a focus of testing of the bicycle riders in the 3D environment on the road situation and not outside of the road. Other similarities which can be observed are for instance roadside signs, road signs and the road situation, such as the road and bicycle lane widths, curves and angles when intersecting with other roads.

It is also useful to look at the real life situation and the 3D model from a top down view, so that the curves, angles and the general outline of the roads can be better compared, as shown in figure 3.24 and 3.25.

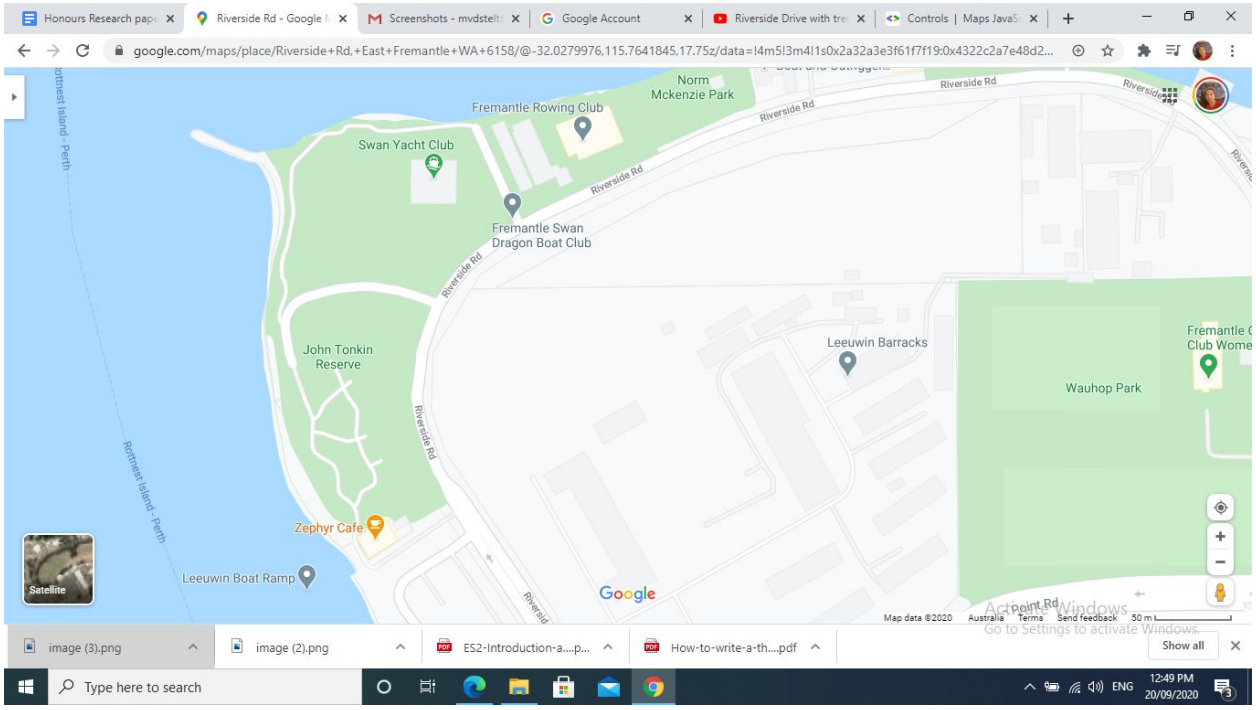

Figure 3.24, Riverside Road top view from Google Maps screenshot, (Machiel van der Stelt, 2020).

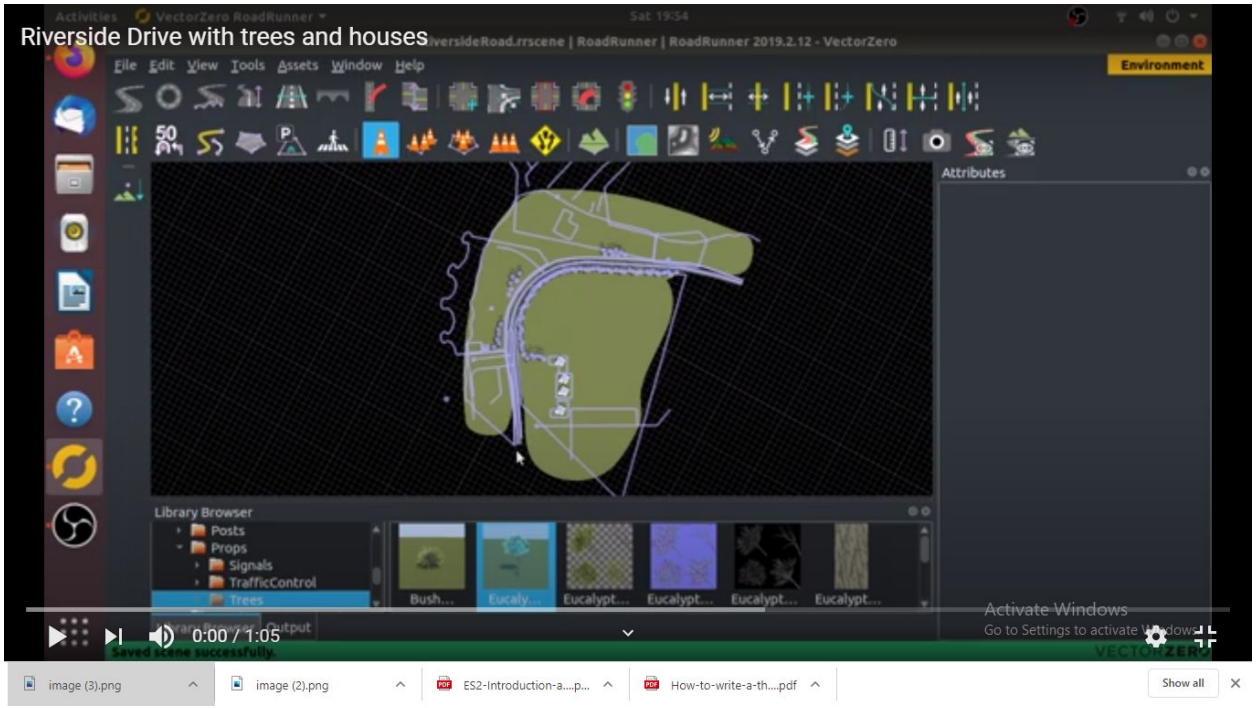

Figure 3.25, Riverside Road top view 3D model in RoadRunner, (Machiel van der Stelt, 2020).

As can be seen, both roads in the Google maps view and the in 3D model RoadRunner look very similar, in terms of curvature of the road. The comparisons made between figures 25 and 26 and the comparisons made between figures 27 and 28 show that the 3D models come very close to the real life situations; this will help the researchers to collect as accurate as possible data, as these two real life locations were specifically selected for their suitability for this study.

Inside Unity3D the functionality of measuring the distance between the Bicycle and the line on the curb on the side of the road needs to be added. It appears that the best way to do this, is to use the built-in functionality of measuring the distance between two objects. This functionality only measures the distance between two objects from both their centre points. The two objects are the bicycle and the line on the curb along the outside of the road. The long object stretching along the curb on the side of the road, the slab of curb, can therefore not be used. This is because only the distance between the bicycle and one point in the long object on the curb will be measured, while we want to measure as many points as possible along the road with the bicycle. The distance between the bicycle and one point on the line perpendicular to the bicycle needed to be measured. This method of measuring the distance from a point on the line to the bicycle. At the same time also the distance of the bicycle to the road line between the bicycle lane and the road for the cars can be easily derived.
To measure the distance between the bicycle and the curb at many as possible points, the best method to do this is by implemented many small poles on the curb on the outside of the road After that is done a C# script needs to be written in a way that the distance is measured only between the bicycle and the closest pole on the curb.
In figure 3.26 a screenshot of the bicycle can be seen with just right of it one pole on Hackett Drive as an example. Many of these poles will be put next to each other at close proximity on

the curb, and made transparent so as not to interfere with the graphics of the virtual 3D environment representations.

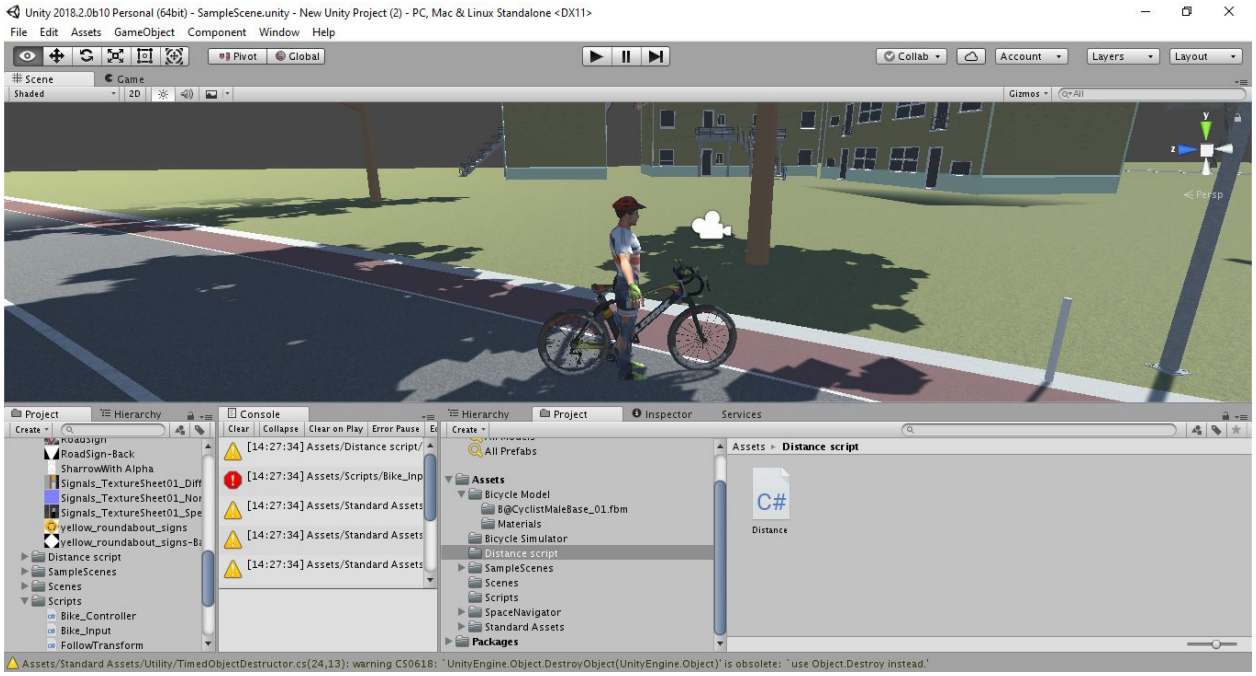

Figure 3.26, The Bicycle on Hackett Drive with just right of it a small measuring pole, (Machiel van der Stelt, 2020).

In figure 3.27 a screenshot of a early version of the C# script can be seen. What needs to be pointed out is that the script starts at line 19, on line 19 the basic Vector3.Distance function built-in for measuring the distance between two objects is called for. After the Vector3 statement both variables ‘Line’ and ‘Bicycle’ can be seen (which are declared earlier in the program as variables) and which are used to measure the distance between them through the Vector3.Distance function. The transform statement used both by the Variables Line and Bicycle, has information stored in regards to the location and other information of that object. The location statement, used after transform for both the Line and Bicycle, was put there to indicate that at this point in time there is only interest in the location information.
For calculating the distance between two objects the Vector3.Distance function first distracts the x, y and z coordinates numbers of both objects. As the bicycle stays at the same height, because there is no elevation applied to the roads, the z coordinates for both (all objects when more poles are used) objects stay the same, with a result of 0 after subtracting. The only variables which then change, according to the movement of the bicycle, are the x and y coordinates. Using the Phytagoras, a Cos or a Sin rule (which is built in the Vector3.Distance) the straight line distance is calculated between the objects Bicycle and Line.
Line 20 in the script as shown in figure 3.27 is a test with a ‘if’ condition. The ‘if’ condition will return a TRUE boolean statement when the distance between the objects is bigger than 1.2 metres.

In the case that the distance is bigger than 1.2 metres, and the return of TRUE boolean of the ‘if’ statement is given, this means that the bicyclist has crossed the road line between the Bicycle lane and the road for the cars and is positioned or partially positioned on the road for the cars. Line 22 in the script as shown in figure 3.27 shows that if the Boolean is returned as TRUE in regards to the ‘if’ statement just described above, the distance in centimeters the bicycle has crossed over the line onto the road is printed onto the inspector screen in Unity3D.

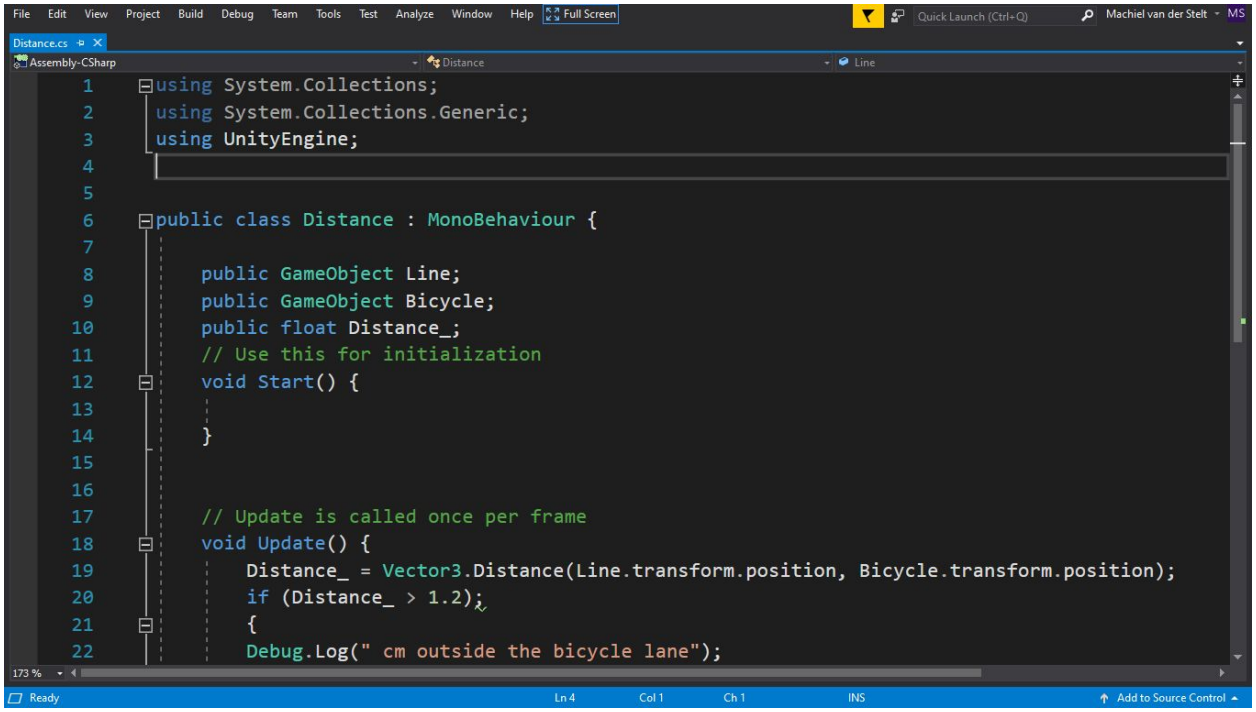

Figure 3.27, Early script for measuring distance between bicycle and a measuring pole, (Machiel van der Stelt, 2020).

To explain the measuring of the distance between the bicycle and the determining the pole with the shortest distance to the bicycle, figures 3.28 and 3.29 were created. Looking at both figures, there can be seen that the green on the left which represents the grass area next to the road, then right from the grass, the dark red area, the bicycle lane is shown and right from the bicycle lane the grey road for the cars is pictured. Between the road for the cars and the bicycle lane, the narrow light grey line is placed, which represents the road line, while the dotted grey and white line between the grass and the bicycle lane is the curb with the small grey squares representing each distance measuring pole. The blue circle on the bicycle path in each figure 3.28 and 3.29, indicate the location of the bicycle.
In figure 3.28 is shown that the bicycle is about half way along the bicycle lane with a number of lines between the bicycle and the distance measuring poles. The C# script will be written in a way that it quickly scans the distance between the bicycle and all the distance measuring poles. The function to determine shortest distance between bicycle and a distance measuring pole among all distance measuring pole will be built-in the C# scripts. The shortest distance between

the bicycle and any of the distance measuring poles is indicated as a solid line, while the other poles with a longer distance to the bicycle are indicated with a dashed line.
As the bicycle movies along downward for instance as shown in figure 24, the C# script keeps on running and constantly determining the new shortest distance between the bicycle and the next distance measuring pole.
Another point which needs to be taken into account is that the bicycle will always be between two distance measuring poles, where the distance between a distance measuring pole and the next distance measuring pole will be equal. This is important to realize. So the script can be adjusted to this, to prevent that the generated data will be corrupted with double values. While it can be observed in the figures that the dashed lines are limited to a number of distance measuring poles, in reality the software will measure the distance between all distance measuring poles and the bicycle, so one line should be solid and all the rest of the lines should be dashed.
This technique needs to be applied to all 4 sections, the Hackett Drive with and without treatment and Riverside Road with and without the wider bicycle lane.
When these 4 sections are rearranged into the 4 different routes it should not affect the accuracy or quality of the data. It means that the poles are scanners in a different order along the whole stretch of each of the 4 trajectories. This will be especially no issue when the scanning is fast enough; but the speed of scanning is also important for the data quality when the scanning is done along the stretch of the bicycle lane in the ‘right’ order.

At every end of each of the 4 routes there would be a void, as each of the 4 sections are designed to seamlessly fit to any other section and when not connected to another section, the road would abruptly stop. To prevent this cosmetic issue a general roundabout that fits on the end of each of the 4 routes was developed, and if needed the bicyclist can bicycle back to the beginning, but the way back the data should ignored as it then starts to measure the distance between the distance measuring poles and the bicycle, while the bicycle is on the other side of the road.
The distance of the Hackett Drive is approximately 291 meters, while the distance of Riverside Road is approximately 622 meters. As both sections are used twice (with and without treatment) in each of the 4 routes, the distance of the total route for each of the 4 combinations will be 2 times 291 meters plus two times 622 meters, which equals approximately 1826 meters. Add to this the short stretch of practice road before the start of the whole route. Then each test subject who participates in the bicycle safety study will have to bicycle about 1850 meters to finish their test and in order to gather enough data.

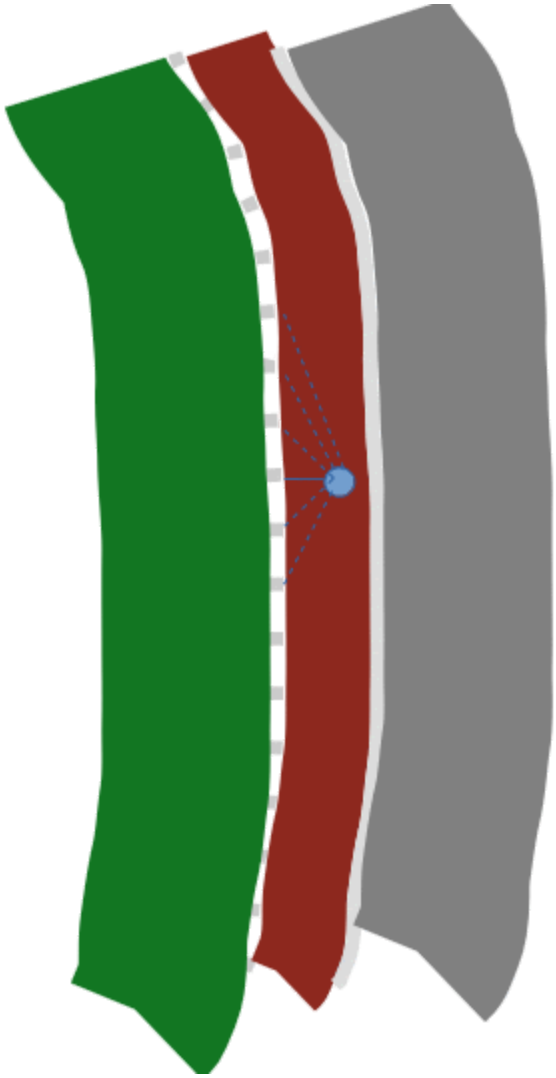

Figure 3.28 Measuring the bicycle distance, (Machiel van der Stelt, 2020).

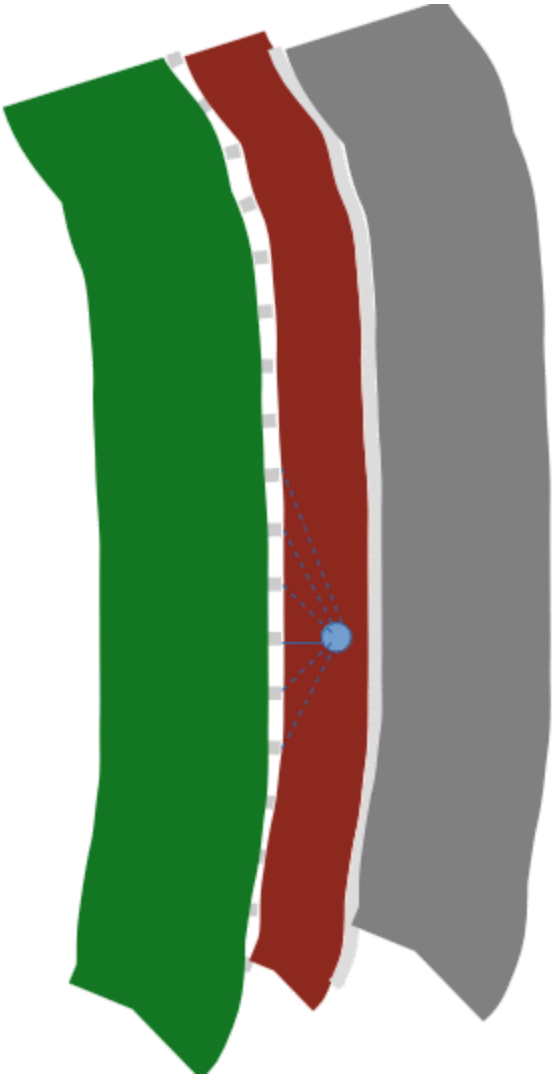

Figure 3.29 Measuring the bicycle distance, bicycle at a different position than in figure 3.28, (Machiel van der Stelt, 2020).

RoadRunner was used to build the 3D models as it is specially designed to model road infrastructure and its surroundings. It has many built-in objects which can be selected and dropped on the places where they need to be used. However some objects needed to be created as they didn't exist in the software library or the shape was not right.
A number of examples of these issues are Roundabout signs with arrows going the opposite direction, no presence of the yellow roundabout signs and wrongly shaped road Island which were automatically generated by the software.

While RoadRunner is designed to quickly select and drop objects onto the plane to create a 3D environment, such as stretches of road and roundabouts; RoadRunner is also designed in a way that these different objects easily snap together. This works well when, for instance,

designing a symmetric roundabout with roads connected to it, but becomes problematic when there is a need for an asymmetric road, such as roads coming in at different angles into the roundabout or if opposing roads have different widths.
One way to solve this is to use the slice tool and slice the part of the road on both sides so it can be adjusted independently from the rest from the traject.
The slice tool needs to be used carefully as the slicing of roads cannot be reversed, apart from the undo option. If in later stages the other edits need to be applied, the sliced part of the road then still needs to be edited separately, which in some cases may be problematic as the sliced part of the road acts differently in the editing process than the rest of the road.

While RoadRunner is a good modelling tool for infrastructure with or without traffic, it is less suitable to be used as a game creator or game console, like when interacting with external equipment to move objects or a single object in the whole 3D environment are required. If interested in using a 3D environment in a gaming setting, then Unity3D is a good and free software option to do that. This is a wide accepted and software application with a very active online help forum.
The bicycle simulator study needs a gaming environment, because the bicycle needs to be able to efficiently communicate with the computer, so that events happening on the bicycle are correctly and efficiently translated and interpreted for use in the 3D model, which runs in Unity3D.

Only inside software packages such as RoadRunner and Unity3D, the 3D models can be put in one bigger 3D model and interact with each other. For this reason it is not possible to export moving traffic in the 3D model to any other 3D modelling software.
Traffic and interaction between 3D models, which need to be used in Unity3D, need to be then created in Unity3D.
As mentioned before very useful feature in Unity3D is the function or ability to measure the distance between objects, this is done by writing a C# script in Unity3D. For the bicycle simulator research project it's important to have hard data showing the location of the bicycle in the 3D environmental model so that can be determined what the impact is of the different road treatments.
Because the written C# script only generates data in relation to the distance between two objects and not the location of the moving objects, the decision was made to also trace the moving object graphically with a line within the 3D environmental model. Since the 3D bicycle model doesn't move in a straight line and location data only is generated from the perspective of the total plane in Unity3D, it's almost impossible to intrepid hard total plane location data in terms of the location of the Bicycle 3D model in the 3D environmental model.
As discussed earlier in this paper, with the method of using the function of measuring the distance between objects to determine the location of the bicycle in relation to the bicycle lane it is best to put transparent poles along the road on the curb with a minimal distance. As the bicycle moves along the road and passes one pole after another the distance gets measured between the bicycle and the shortest distance between the bicycle and pole gets recorded and stored in a spreadsheet.

Also during a meeting with the researchers it was determined that the point on the bicycle, which is used for the measurement of the distance between the bicycle and the poles is the central point of gravity in the middle of the bicycle.

4 Conclusions

First point to make is that hardware and software technologies have improved over the last two to three decades, which made it possible to reproduce more accurate 3D and virtual environments. Compared with technologies from the 1970's, current hardware and software technologies are able to create more realistic virtual environments. Current technologies also have the ability to collect more diverse and accurate data. For instance in the 1970's the collection of data was limited to mainly impulse and reaction data of the test subject. Current technologies however can at least collect location, speed, collision and directional data of the object of interest.

While software packages have a great library of objects to add to the virtual environment, the demand for other objects or special of users often will stretch beyond many libraries. There are possibilities to 3D scan reaL life objects which need to be used, with relatively accessible technologies. For instance there exist some paid and unpaid Apps for mobile phones, but with the limited computing power of smartphones for this relatively new method called Photogrammetry, the results are limited and possibilities of the diversity of 3D objects are limited. However there exists one free software which can be run easily on Desktop computers and possibly laptops with a NVIDIA Graphical Processing Unit (GPU), This software, called Meshroom, can produce surprisingly good results in creating virtual 3D objects scanned through taking 30 or more pictures around the object in a circular shape, if the object of interest isn't shiny or reflective. However importing these scanned 3D models may be an issue for 3D modelling software either for technical reasons such as that the 3D file format is not compatible resulting in the 3D object looking different than how it looked in Mushroom. Also, if the object doesn't have the desired dimensions for the virtual environment, and scaling needs to be done disproportionate into the three dimensions, this will result in a misshaped 3D object.

Nowadays there is available a lot of accurate and reliable spatial data, which helps to reproduce virtual environments more reliably and quicker. While there are different kinds of quality (expressed in detail) spatial data are available, either freely or commercially, the free basic vector data from OpenStreetMap (OSM) proved to be a reliable and accurate source to create a representative 3D virtual environment.
Software and hardware technologies have come to a stage where they are very sophisticated, and which enables users such as researchers to create very realistic 3D environments. This is mainly due to the increased graphics capability. The increased computing capability made it possible to collect accurate and diverse data.

The added advantage of using a virtual environment is that a safe environment can be created to perform for instance road safety studies, in addition to the possibility to adjust the 3D environment which in the real world would be too expensive to difficult.

For small and very specific 3D environment requirements, it is advisable to use simple spatial data such as from OpenStreetMap. With this simple spatial data 3D environments can be built

exactly to the requirements. Ready built 3D environments could be too difficult to adjust to the specific requirements.
Scanning of 3D objects is not advisable as the 3D objects may not fit well in the 3D environment both in terms of graphics, and dimensions.
Current technologies enable the creation of safe and flexible virtual 3D environments for road safety studies. However because of limitations they can not exactly replicate the real life situation. Some examples of these limitations are for instance some physical components from real life are missing, and the absence of random events. For this reason road safety studies with a virtual 3D environment can be used for a limited amount of measuring points.
This research has shown that with relatively limited resources, a reliable and accurate 3D environment can be built for road safety studies to collect reliable and detailed data. The RoadRunner software application is well suited to design and build a wide variety and specific 3D environments, while Unity3D is a good solution to create a player environment and a scripting ability, which is flexible enough to collect a wide variety of data.

In order to make the 3D road testing environment more representable of the real life situations, the following points could be considered. The creation of testing systems with a better integration of both virtual 3D environments and physical phenomena from real life. Another consideration could be to implement testing equipment in vehicles or on bicycles in real life situations in combination with the 3D virtual environment testing.

5 References

1. Western Australian Road Safety Commission, 2020
   https://www.rsc.wa.gov.au/

2. Western Australian State Government passing new passing-distance laws, 2017
   https://www.rsc.wa.gov.au/Rules-Penalties/Browse/Cyclists

3. Reported Road Crashes: in Western Australia 2002, RSC, 2002
   https://www.rsc.wa.gov.au/RSC/media/Documents/Road%20Data/Statistics/Annual%20crash%20statistics/annual-crash-statistics-2002.pdf

4. 2015 Reported Road Crashes in Western Australia, RSC, 2015
   https://www.rsc.wa.gov.au/RSC/media/Documents/Road%20Data/Statistics/Annual%20crash%20statistics/annual-crash-statistics-2015.pdf

5. Scenarios and setting description by Michelle Fraser of the School of Public Health at UWA, 2020

6. 'Safer Cycling and the Urban Road Environment', Meuleners L, Fraser M, November 2018

7. 'Validation of a bicycle simulator for road safety research', Steve O'Hern a, Jennie Oxley, Mark Stevenson. 2017

8. Unity3D explained on Wikipedia, 2020
   https://en.wikipedia.org/wiki/Unity_(game_engine)

9. Hackett Drive location on OpenStreetmaps, 2020:
   https://openstreetbrowser.org/#buildings-height/w50431889&map=19/-31.98543/115.82187&categories=leisure,roads,buildings-type,buildings-height

10. Riverside Road on OpenStreetmaps, 2020:
    https://openstreetbrowser.org/#buildings-height/w50431889&map=18/-32.02830/115.76460&categories=buildings-height,leisure,roads,buildings-type

11. 'A validation study of driving errors using a driving simulator:', (Lynn Meuleners, Michelle Fraser, 2015).
    https://www.sciencedirect.com/science/article/abs/pii/S1369847814001764

12. ‘Can driving simulation be used to predict changes in real-world crash risk?’, (Christina M. Rudin-Brown, Amy Williamson & Michael G. Lenné, 2009).

https://www.monash.edu/muarc/archive/our-publications/reports/muarc299

13. ‘Driving simulator: A VR tool to help iexperienced driver to undestand with the road infrastructure.’, (Mohammad Yanuar Rizki, 2019) (sic)

https://www.academia.edu/38685332/DRIVING_SIMULATOR_RESEARCH

14. VectorZero RoadRunner user documentation, (VectorZero,Mathworks, 2020).

https://www.mathworks.com/help/roadrunner/index.html

15. ‘Alcohol Effects on Driving Behavior and Performance in a Car Simulator’, (R. Wade Allen, Henry R. Jex ; Duane T. McRuer ; Richard J. Dimarco, 1975).

https://ieeexplore.ieee.org/abstract/document/5408372

16. ‘Traffic modeling software for IVHS applications’, (A.J. Santiago ; H. Chen, 1990).

https://repository.lib.ncsu.edu/bitstream/handle/1840.4/5837/1990_0133.pdf?sequence=1

17. ‘Virtual environment construction for driving simulator’, (T. Yasuda ; T. Suzuki ; S. Yokoi ; J. Toriwaki, 1994).

https://ieeexplore.ieee.org/abstract/document/365957

18. ‘Vehicle system simulation based on oriented-object and visual modeling’, (Cheng Jun ; Gao Yuekui ; Wang Tao, 1999).

https://ieeexplore.ieee.org/document/830701

19. ‘Road hazard reaction testing using driving simulation: the novice vs. the experienced drivers’, (Y. Wang ; P. Peng ; Lijun Liang ; W. Zhang ; S. Wu, 2008).

https://ieeexplore.ieee.org/abstract/document/4419235

20. ‘Influence of in-vehicle music on driving: experimental results with a driving simulator’, (Keisuke Mizoguchi ; Sadayuki Tsugawa, 2012).

https://ieeexplore.ieee.org/abstract/document/6294296

21. Crimson workstation specifications, (Wikipedia, 2020).

https://en.wikipedia.org/wiki/SGI_Crimson

22. Object Oriented programming, (Wikipedia, 2020).

https://en.wikipedia.org/wiki/Object-oriented_programming

23. Object Oriented programming, (Programming with Mosh, YouTube, 2018).

https://www.youtube.com/watch?v=pTB0EiLXUC8

24. Object Oriented programming, (Studytonight, YouTube, 2018).

https://www.youtube.com/watch?v=xoL6WvCARJY

25. Meshroom 3D Photogrammetry software application main website, (AliceVision, 2020)

https://alicevision.org/

26. Meshroom tutorial 1, (Gamefromscratch, YouTube, 2018).

https://www.youtube.com/watch?v=R0PDCp0QF1o

27. Meshroom tutorial 2, (Sketchfab, 2019).

https://sketchfab.com/blogs/community/tutorial-meshroom-for-beginners/

28. Polygon Mesh Model of a Dolphin, (Wikipedia, 2020).
https://en.wikipedia.org/wiki/Polygon_mesh#/media/File:Dolphin_triangle_mesh.png

29. Polygon Mesh models, (Wikipedia, 2020).

https://en.wikipedia.org/wiki/Polygon_mesh

30. Photogrammetry technique for 3D photo realistic models, (Wikipedia, 2020).

https://en.wikipedia.org/wiki/Photogrammetry

31. Calculation of distance between two objects in Unity3D, (BeepBoopIndie, YouTube, 2017).

https://www.youtube.com/watch?reload=9&v=OMPV-duv25Q

32. Distance between multiple objects in Unity3D, (DitzelGames, YouTube, 2018).

https://www.youtube.com/watch?v=6rDlfYC4HxM

33. Using the Prefab method to create multiple objects for effective distance measuring, Unity, (Unity Technologies, 2020).

    https://docs.unity3d.com/Manual/CreatingPrefabs.html

34. K-D trees in Unity3D to determine the shortest distance between many objects, (Wikipedia, 2020).

    https://en.wikipedia.org/wiki/K-d_tree

35. Closest distance of one player object to many other objects in Unity3D, (Indie Games Dev, YouTube, 2020).

    https://www.youtube.com/watch?v=VH-bUST_w0o