RoSaVR: IMMERSIVE VR APPS FOR ROAD SAFETY FOR KIDS

RAMAN BIN ARAS

CD12045

BACHELOR OF COMPUTER SCIENCE (GRAPHICS & MULTIMEDIA TECHNOLOGY) WITH HONOURS

MISS NORANIZA BINTI SAMAT

FACULTY OF COMPUTER SYSTEMS & SOFTWARE ENGINEERING (FSKKP) UNIVERSITI MALAYSIA PAHANG

RoSaVR: IMMERSIVE VR APPS FOR ROAD SAFETY FOR KIDS

RAMAN BIN ARAS

A thesis submitted in fulfilment of the requirements for the award of the
Degree of Computer (Graphic and Multimedia Technology)

Faculty of Computer Systems & Software Engineering
Universiti Malaysia Pahang

MAY 2016

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's full name : Raman Bin Aras

Date of birth : 02 January 1992

Title : RoSaVR: Immersive VR apps for road safety for kids

I declare that this thesis is classified as:

□ CONFIDENTIAL (Contains confidential information under the Official Secret Act 1972)

□ RESTRICTED (Contains restricted information as specified by the organization where research was done)

□ OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserve the right as follows:

1. The thesis is the Property of Universiti Malaysia Pahang.

2. The library of Universiti Malaysia Pahang has the right to make copies for the purpose of research only.

3. The library has be the right to make copies of the thesis for academic exchange.

Certified by:

------------------------	------------------------
(Student's Signature)	(Signature of Supervisor)
931211-06-5388	Miss Noraniza Binti Samat
Date: 12 January 2016	Date: 12 January 2016

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries, which have been duly acknowledged.

Signature	: …………………………
Student Name	: Raman bin Aras
Date	: 18 May 2016

SUPERVISOR’S DECLARATION

I hereby declare that I have read this thesis and in my opinion this thesis/report is sufficient in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Graphic and Multimedia Technology)

Signature : ……………………………….……..
Supervisor Name : Miss Noraniza Binti Samat
Date : 18 May 2016

ACKNOWLEDGEMENTS

I would like to express my very deep gratitude to Allah S.W.T, the Almighty God for the grant me the strength to complete my Final Year Project II for this semester. I am very honored and grateful to Miss Noraniza Binti Samat for being my supervisor. She advises and comments have been a great contribution for the achievement of my thesis and the objectives of this project.

My appreciation also goes to my Academic Advisor, Dr. Ngahzaifa Binti Ab Ghani and my fellow lectures who had taught me along these semesters in Universiti Malaysia Pahang.

Besides, not forget to mention, a huge appreciations to my beloved parent and family especially to my mother Nurhadiah binti Hammade who are always supporting and encourage me throughout my years of study.

Then, to all my fellow friends specifically to Legend Six and Joyah Team for spending their precious time on support me during this project.

ABSTRACT

The term “Virtual Reality” is very widespread nowadays. It is because of the development in technology especially the smart phone technology and the computer technology. Back then, developing virtual reality system takes a lot of time and money and that includes many men power in the development process. Nowadays, the virtual reality technology becomes chip in the term of hardware and the software in development process, because of that many people try to develop virtual reality in many aspect. For example, the training simulation for pilots, treatment for person that has phobia in darkness or animals such as spider, dogs and cats. Moreover, virtual reality also becomes more popular in computer games technology, which was installed into their smart phone. They can get into the game only by using the smart phone that was placed into the Google. So based on the technology that we have today, this project is using virtual reality technology in learning process for the children to learn about road safety and help them to increase their understanding in that matter. The laws from the book are applied into this project to make a change from reading to immersion to the situation of every road law. So this project can increase children’s understanding on road safety rather than only reading the book. Lastly, the biggest hope that this project can help and achieve the objectives of this project.

ABSTRAK

Istilah "Realiti Maya" digunakan sangat meluas pada masa kini. Hal ini disebabkan kerana, berkembangnya teknologi terutamanya teknologi telefon pintar dan teknologi komputer. Pada masa dahulu, membangunkan sistem realiti maya mengambil banyak masa dan wang dan dan juga termasuk penggunaan tenaga yang ramai. Pada masa kini, teknologi realiti maya menjadi lebih murah dalam bentuk alatan dan perisian dalam proses pembangunan,oleh itu ramai, orang cuba untuk membangunkan realiti maya dalam banyak aspek. Sebagai contoh, simulasi latihan untuk juruterbang, rawatan untuk orang yang mempunyai fobia dalam kegelapan atau haiwan seperti labah-labah, anjing dan kucing. Selain itu, realiti maya juga menjadi lebih popular dalam teknologi permainan komputer, yang telah dipasang ke dalam telefon pintar mereka. Mereka boleh masuk ke dalam permainan hanya dengan menggunakan telefon pintar yang diletakkan ke dalam Google. Jadi berdasarkan kepada teknologi yang kita ada hari ini, projek ini menggunakan teknologi realiti maya dalam proses pembelajaran untuk kanak-kanak untuk belajar tentang keselamatan jalan raya dan membantu mereka untuk meningkatkan pemahaman mereka dalam perkara itu. Undang-undang dari buku digunakan dalam projek ini untuk membuat perubahan dari membaca kepada berhubung terus kedalam setiap undang-undang jalan raya. Jadi projek ini dapat meningkatkan pemahaman kanak-kanak mengenai keselamatan jalan raya dan bukan hanya membaca buku. Akhir sekali, diharapkan projek ini dapat membantu dan mencapai objektif projek ini.

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF TABLE

LIST OF ABBREVIATION/ACRONYM

OBBREVIATION	**TITLE**
VR	Virtual Reality
VE	Virtual Environments
HMD	Helmet-Mounted Display
3D	Three Dimension
WHO	World Health Organization
IMU	Inertial Measurement Unit
SDLC	Systems Development Life Cycle
RAD	Rapid Application Development
IDE	Integrated Development Environment
SDK	Software Development Kit

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Virtual reality (VR) or Virtual Environments (VE) is designed to emulate the look and feel of being in real environments that user can immerse into an artificial computer-simulated environment. Importantly, virtual environment afford actions - which can form the basis of experimental learning. Virtual environments are rendered on a computer screen, projected onto external surfaces, or experienced with a head mounted display, permitting the user to move about freely in a virtual environment (Waller et al., 2007).

In the used of virtual reality the US military has used virtual reality to train users how to fly planes and simulate how it would feel to jump from a plane. The creation of large-scale virtual environments permits users to engage in (and researchers to study) complex full motor behaviours such as fielding a batted baseball (Fink et al., 2009). Virtual reality helps in training and simulation that requires a lot of time and money to be spent for cheaper and successfully training. In terms of interaction with virtual environments, users can rely on classical devices such as the computer mouse, joystick, and gamepad to act in the virtual space (James

D. et al., 2015). The device can send the data interaction onto the virtual system that enables users freely interact with the system.

Head-mounted display (or helmet-mounted display, for aviation applications), both abbreviated HMD, is a display device, worn on the head or as part of a helmet that has a small display optic in front of one (monocular HMD) or each eye (binocular HMD). A typical HMD has either one or two small displays with lenses and semi-transparent mirrors embedded in a helmet, eyeglasses (also known as data glasses) or visor. In this project the VR is focusing on the HMD, VR headsets or Oculus rift. By using the smartphone that has VR application and attach the smartphone to the VR headsets and the VR application can be seen thought the smartphone screen.

Road is a long, narrow stretch with a smoothed or paved surface, made for travelling by motor vehicle, carriage, etc., between two or more points; street or highway (Christine Ammer, 2002). Roads are used to connect one place and another so that the travelling time is faster and smoother. However, because of all people using the same road, the road safety was enforced and it applies on the roads and all the users must follow the laws to make the road safe. The road safety enforcement is used in enforcing the laws that control the use of motor vehicles and the behaviour of drivers, cyclists, pedestrians and other road users.

In this project, the combination of VR and road safety is applied to make a learning aid for the children to give extra understanding about the road safety enforcement. So, using these applications children can learn two things about VR technology and also road safety enforcement.

1.2 OBJECTIVES

The objectives of this project are:

1. To study about the use of Virtual Reality in learning (Road safety).
2. To design and develop Virtual Reality mobile application for road safety.
3. To evaluate the effectiveness of learning road safety using VR application.

1.3 SCOPES

The scopes of this project are:

1. Virtual reality technology using mobile application with virtual reality box and game controller.
2. Learning aid for road safety for kids (6 - 12 year old).
3. Fully interactive 3D VR learning environment.

1.4 PROBLEM STATEMENT

Due to rapid development of smartphone and technology in our country, Virtual Reality seems to be a great platform to be used in learning to increase understanding and performances about road safety that can reduce costs, times and others. Basically, during class, learning processes would be based on books and explanation from teachers. A good student will follow the steps in books correctly and understand the road safety. However, not all students can do that. They need some video tutorials. But the videos only show two-dimensional. Teachers are still using the old method, which is using books and illustrations. By referring to books it will take a long time to study the road safety.

Besides, today many accidents happened on the road that includes pedestrians and also children. This happened because some of the drivers and also the children do not know the road safety. Thus, by implementing this project the problems can be solved to change or add the new method and use Virtual reality for fast learning and better understanding of each law. Furthermore, by carrying out this project, teacher can give lectures to new students using virtual reality for them to understand better through the usage of virtual object in three-dimensional objects that has in the application. Finally, this project will help to give a better understanding.

1.5 THESIS ORGANIZATION

This thesis consists of five chapters. Chapter one will discuss on introduction to system or research. Chapter two will talk about literature review, to explain about the selected project. This chapter is divided into two sub-researches, which are Existing Research or system and Technique or hardware.

Chapter three will discuss on methodology about overall approach and framework of research. Chapter three also propose develop framework and model through the workflow. Chapter four is the implementation process; it describes how the research was structured. Chapter five will discuss and explain about the finding or result from the data analysis. Lastly, is to make conclusion for research that had been done.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter briefly describes the review on existing techniques related with road safety using virtual reality that will be developed later. This chapter also reviews the existing techniques, which is implementing VR in any system. This chapter consists of two partitions. The first partition describes the comprehensive review on existing related system. The second partition describes the review on method, hardware, and the technology previously used in the same domain.

2.2 ROAD SAFETY

Road traffic crashes are one of the world's largest injury prevention and public health problems. According to the World Health Organization (WHO), more than 1 million people are killed on the world's roads each year (Peden, M., et al

2004). A report published by the WHO in 2004 estimated that some 1.2 million people were killed and 50 million injured in traffic collisions on the roads around the world each year (Peden, M., et al 2004) and was the leading cause of death among children 10–19 years of age. The report also noted that the problem was most severe in developing countries and that simple prevention measures could halve the number of deaths (BBC News, 2008).

The 2013 World Health Organization (WHO) report that purportedly showed that Malaysia has the highest deaths on the road compared to any other nation in the world. Malaysia’s road fatalities currently stand at 25 deaths per 100,000 populations, a value which is higher than India (19.9), Russia (18.6), and China (20.5) (Peden, M., et al 2004). In 2005, Al Haji from the Linköping University, Sweden measured the road safety levels of ten ASEAN countries from 1994 to 2003 and found that the road safety levels among these countries differed widely from one another. Malaysia in particular was ranked third for having the safest roads among the ASEAN countries, but ranked far ahead of us at the first and second positions were Singapore and Brunei, respectively. Al Haji also found that Singapore and Brunei also had similar road safety level with Sweden’s, a highly developed nation. In contrast, Myanmar, Vietnam, Cambodia, and Laos were determined to have the least safe roads among the ASEAN countries (Al Haji, G. 2005).

2.2.1 Existing system for Road safety Learning Aid

A. Book

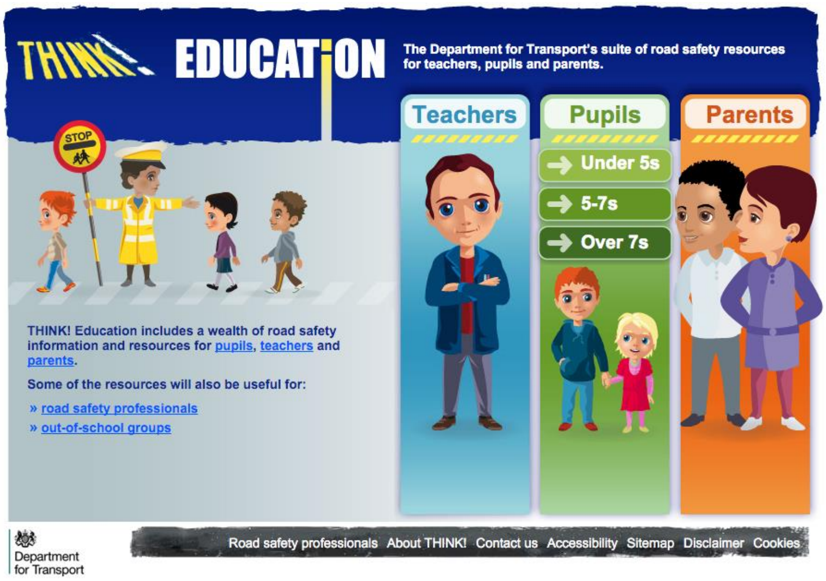
Book is defined as a set of written or printed-paper that compiled to hinge at one side. Book usually consists of texts, diagrams and pictures. Books about road safety are one of the most important references in studying the road safety and road traffic. Books that are used in teaching are also a medium of reading. The safety ways, types of road and instruction of a road safety are stated in the book. However, the usage of books in teaching can also cause some difficulties for students to understand the contents and for lecturers to explain the topics. Bellow shows the example book that used to teach parent and kids about road safety. (Source: http://www.kidsandtraffic.mq.edu.au/)

Gazette
Safe Journeys Safe Communities
NOW OPEN
SAFE JOURNEYS SAFE COMMUNITIES
When children are out and about hold their hands to keep them safe.
KIDS and TRAFFIC

Figure 2.1: Safe Journeys Safe Communities book

B. Web Educations

Web education is example web that teach teacher, pupils and parents about road safety. For example, THINK! provides road safety information for road users. Their aim is to encourage safer behaviour to reduce the number of people killed and injured on the road every year. (Source: http://think.direct.gov.uk).



The Department for Transport's suite of road safety resources for teachers, pupils and parents.

THINK! Education includes a wealth of road safety information and resources for pupils, teachers and parents.

Some of the resources will also be useful for:

» road safety professionals
» out-of-school groups

Figure 2.2: Web educations that provided by

Department for Transport in UK

C. Mobile Application

A mobile app is a software application developed specifically to be used on small, wireless computing devices, such as smartphones and tablets, rather than desktop or laptop computers. Mobile apps are designed with consideration of the demands and constraints of the devices and also to take advantage of any specialized capabilities they have. A gaming app, for example, might take advantage of the iPhone's accelerometer. Nowadays, there are many mobile applications that teach children about road safety. Example,

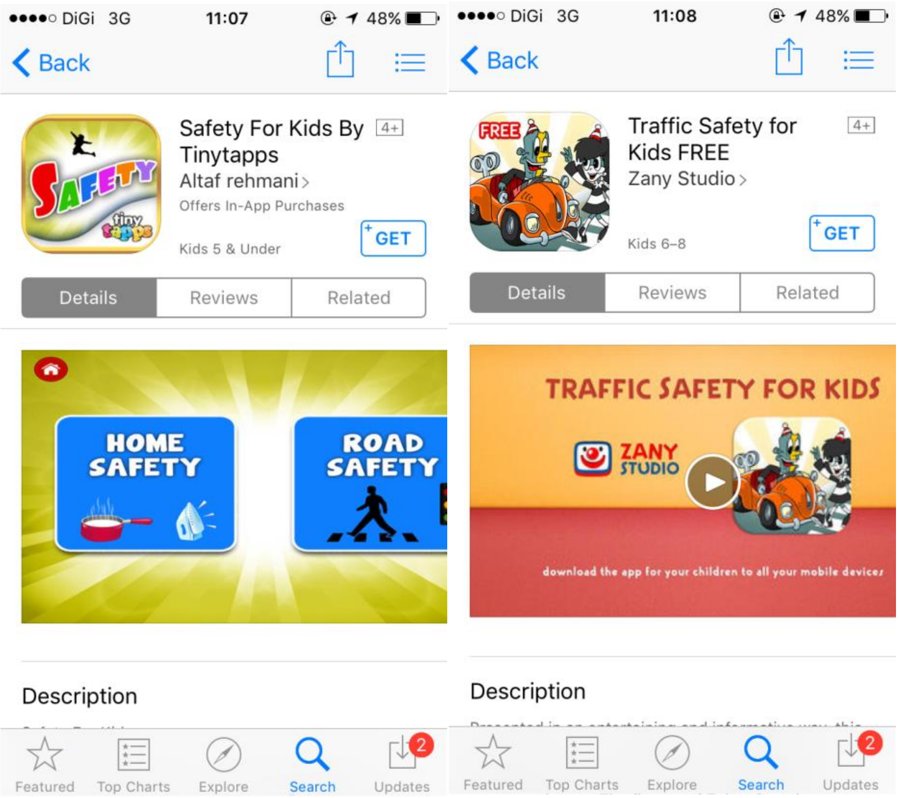


Figure 2.3: Example mobile applications that available at Google play nowadays.

D. 3D Animations

3D computer graphics are graphics that use a three-dimensional representation of geometric data that is stored in the computer for the purposes of performing calculations and rendering 2D images. Such images may be stored for viewing later or displayed in real-time. In this 3D animation the application of visual simulation was designed to allow direct access to any stage of the process in which it is based and may be viewed from any point within the virtual scene, thus facilitating their understanding. (Source: https://www.youtube.com/watch?v=VCMX8BjuOhE)

Figure 2.4: Example 3D animations for safety road education

Figure 2.5: Other Example 3D animation that teach kids about road safety

Features / Criteria	Web Education	Books	3D Animations	Mobile Application	RoSaVR
Point of view	2D	2D	3D	2D	360^{0}
Music applies	Yes	No	Yes	Yes	Yes
System Interactivity	No	No	Yes	Yes	Fully Interactive
Cover the chapter of road safety education	Yes	Yes	No	No	No

Table 2.1: Table of compression existing system with this project

2.3 VIRTUAL REALITY

Virtual reality is an artificial environment that is created with software and presented to the user in such a way that the user suspends belief and accepts it as a real environment. On a computer, virtual reality is primarily experienced through two of the five senses: sight and sound (Margaret Rouse, 2015). Virtual reality or virtual realities (VR), which can be referred to as immersive multimedia or computer-simulated reality (Hu Fei, 2016) replicates an environment that simulates a physical presence in places in the real world or an imagined world, allowing the user to interact in that world.

VR itself is a digital device that is helpful and useful to provide a virtual environment to the human experience. They are example, function and benefits of the virtual reality (Maxmanroe.com, 2016).

1. Field of Media and Entertainment (Games, Simulation Environment)

   Function and use of Virtual Reality is the first for the field of media and entertainment. For the user that lovers of games of course the experience of playing game will become an increasingly significant place.

2. Field of Medical and Medicine (Anatomy of the Human Body)

   VR device, it is also useful in the medical field and medical. With this device, doctors will be able to describe the anatomy of the body (simulating human anatomy). From here the organs in the body will be able to look more real. By knowing the condition of the organs of the body in a more clear then the doctors and surgeons will be able to determine more accurately the decision to do surgery or not. In addition to these benefits, VR can also be helpful for the doctor to see the development of a disease, diagnosing a disease and patient care techniques are more appropriate.

3. Field Military (War)

   In the military field, the use of Virtual Reality will be able to help the soldiers to a war simulation. Like the play Point Blank, the soldiers would get more real combat experience and clear. VR is able to present a more realistic view of the digital battlefield environments and situations that are going to make war soldiers get to experience an amazing and interesting. In this way it will also make the cost of the war games are far more efficient and more economical when compared to actual combat training.

4. Transportation Sector (Commercial Aircraft Simulation and Combat Aircraft)

   Training to fly of course will make us have to incur huge costs. But with the VR device is the one who will be doing the training to fly a plane no longer have to spend big to lease aircraft and fuel. Because the VR you can get experience flying the aircraft a very real and dangerously close to reality. With the use of VR for flight training, you will be protected from things that are worrying in flight.

5. Engineering and Automotive (Car Design)

   In engineering and automotive, VR will be useful for designing a car. One of the companies has used VR technology in the development of the Ford car. To evaluate the outside and inside of cars that have not made it this Ford car designers use this type of headset Oculus Rift. When using the headset, the designers will be able to pay attention to the details of the car. Meanwhile, cameras will follow the movement of the designers and coordinate with the software to match the digital presentations with motion designer.

6. Economic Affairs (Promotional Item Display / Display)

   The usefulness and benefits of VR is to promote goods display or a display. One of the stores that have implemented this way is the Selfridges store on Oxford Street, London. With VR has, visitors can try some of the products sold by the store with more real.

A virtual reality headset is a head-mounted device aimed to provide an immersive virtual reality experience, for the purpose of computer games and 3D simulations. It consists of a stereoscopic head-mounted display (providing separate images for each eye) and head motion tracking sensors (Ben Kuchera, 2016) (which may include gyroscopes, accelerometers, structured light systems, (Adi Robertson)). Some devices also include headphones, eye tracking sensors (Stuart Miles, 2015) and gaming controllers.

2.3.1 Existing system of Mobile

A. Google Cardboard

   The Cardboard project aims at developing inexpensive virtual reality (VR) tools to allow everyone to enjoy VR in a simple, fun, and natural way. The Google developer already made many mobile application in virtual reality to use in Google cardboard and there has 12 mobile application that popular in Google cardboard for example Snow Shaker Maker, Google Cardboard Camera, Rollercoaster and many more. Figure bellow shows the image in Snow Shaker Maker that popular in Google cardboard.

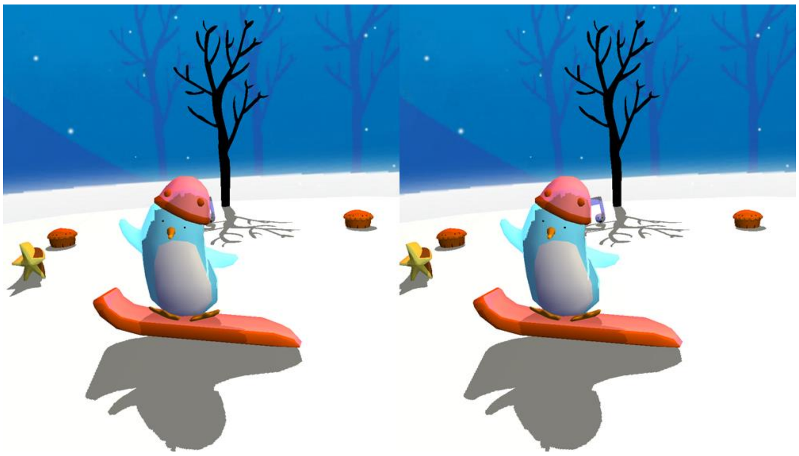


Figure 2.6: Snow Shaker Maker

In Google cardboard user also can build their own application viewer from simple, low-cost components using specifications published by Google, or purchase one manufactured by a third-party. Figure 2 shows the example Google cardboard, simple eco-friendly and confortable for the users.

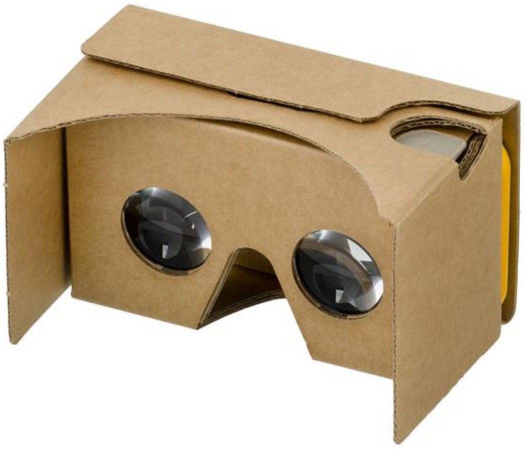


Figure 2.7: Google cardboard

B. Samsung Gear VR

Samsung Gear VR is a mobile virtual reality headset developed by Samsung Electronics in collaboration with Oculus VR. A compatible Samsung Galaxy device (Galaxy Note 5, Galaxy S6/S6 Edge or Galaxy S7/S7 Edge), sold separately, acts as the headset's display and processor, while the Gear VR unit itself contains the high field of view lenses as well as a custom inertial measurement unit (IMU) for rotational tracking which connects to the smartphone via micro-USB. Bellow the example image of Gear VR.

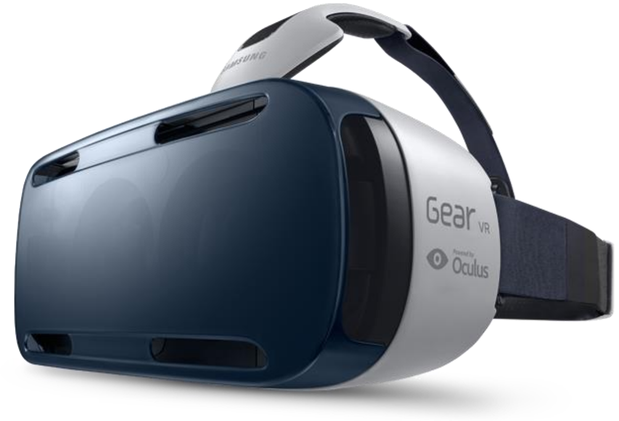


Figure 2.8: Samsung Gear VR

The Samsung Gear VR also needs and develop the mobile application to use it with the Gear VR for example the famous application is Land’s End, Gone, Samsung Internet, and etcetera.

Figure 2.9: Mobile applications for Samsung gear VR

2.4 PROBLEM EXISTING SYSTEM

Every existing system has its own purposes. Same as this project, this is to develop a virtual reality application. All of the existing systems are good and benefit the people. However, this project aims to combine and improve the existing system in a different field. Google cardboard and Samsung gear VR is a good technology in VR world. This is because to develop and provide VR application it takes large number of price and budget but this two system can make every people can buy and not too expensive. This two has own mobile application in gaming, video, and application. However, there no application that teaching and learning application for children study about traffic in road for their safety. So this project develops for make the application for children lean and study about road safety.

Books are one of the tools used in teaching to help and guide students in understanding the subjects. Nevertheless, not all students like to read. Besides, the price of each books are too expensive for some students to buy them.

3D animations consist of videos and 90% of real images with complete guide for understand the road safety. In 3D animations, all of the complete guide can be seen but 3D animations do not involve interactions with people to immerse and fell the real situation. So the 3D animation is good in giving ideas and gives first impartation about road safety.

2.5 VR DEVELOPMENT TECHNIQUES

A. Develop the Stereoscopic view

To get the view off eyes both sides, the camera stereoscopic must be develop. In Unity we use the camera stereo to get the both eyes view. So, this camera in Unity that use in rendering the application and the view will shows the both side view in mobile if run the application.

B. 3D Object

Several technique will be use to develop the object of Virtual reality. The example object will be create is Car. The car is important component that will development in this application, the technique below will be use to create the smooth 3D object and to create the same virtual object with the real object. Before knowing the ways and techniques to develop 3D object knowing the computer graphic also impotent in this project.

Computer graphic is creation, manipulation and storage of geometric objects modelling and their images rendering and display those images on screens or hardcopy devices. There are several examples of the applications of computer graphic such as computer-aided designs, graphic presentations, computer arts, and etcetera. Nowadays, the world focuses more in developing computer graphic because of its benefits and its potentials in technology world, working environment and in sports. There are many types of techniques in computer graphic such as:

i. Rendering Process

3-D graphic design, rendering is the process of adding shades, colours and laminations to a 2-D or 3-D wireframe in order to create life-like images on a screen. Rendering may be done ahead of time (pre-rendering) or it can be done in on the fly in real time. Real-time rendering is often used for 3-D video games, which require a high level of interactivity with the player. Pre-rendering, which is CPU-intensive but can be used to create more realistic images, is typically used for movie creation.

ii. 3D Modelling

3D modelling or 3D computer graphics are works of graphic arts that are created with the aid of computers and specialized 3D software. There are many great applications for digital 3D modelling, though the video gaming industry is one of the most commonly recognized by many. Designers use various three dimensional modelling methods to create realistic character models that appear on gaming consoles and online. The end product can be either animate or inanimate, but is typically characterized by its life-like look and apparent depth. There are several techniques used to develop a 3D modelling:

- Spline or patch modelling:
  A spline is a curve in 3D space defined by at least two control points. The most common splines used in 3D art are Bezier curves and NURBS. The software Maya has a strong NURBS modelling foundation. Using splines to create a model is perhaps the oldest, most traditional form of 3D modelling available. A cage of splines is created to form a "skeleton" of the object you want to create.

- Box Modelling

  Box modelling is possibly the most popular technique, and bears a lot of resemblance to traditional sculpting. In box modelling, one starts with a primitive usually a cube and begins adding detail by "slicing" the cube into pieces and extending faces of the cube to gradually create.

- Mesh Topology

  Another aspect of proper workflow is creating a model with an ideal 3D mesh topology. Topology optimization is usually associated with creating models used in animation. Models created without topology that flows in a smooth, circular pattern, may not animate correctly, which is why it is important to plan ahead when creating any 3D object that will be used for animation.

The techniques above will be used to create all 3D object and the entire component that use in the application scene from start until the end of the scene. The 3D object will be import from other file into the Unity software because all the 3D object created from another software like Maya and Blander.

iii. Texture Mapping

Texture mapping is a graphic design process in which a two-dimensional (2-D) surface, called a texture map, is “wrapped around” a three-dimensional (3-D) object. Thus, the 3-D object acquires a surface texture similar to that of the 2-D surface. Texture mapping is the electronic equivalent of applying wallpaper, paint, or veneer to a real object.

C. Creating System

After creating the stereo camera and the 3D objects for road safety component. The final technique is creating the application that will save in android based. Thus, the file project will use and run in the phone. With the VR box, the application that created will be seen in VR mood. The user can feel the immersive application using the VR box that place the smart phone inside the VR box.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

In this chapter it will discuss about the methodology that used to develop this project. Methodology is defined as a system of broad principles or rules from which specific methods or procedures may be derived to interpret or solve different problems within the scope of a particular discipline. Unlike an algorithm, a methodology is not a formula but a set of practices. Common methodologies include waterfall, prototyping, iterative and incremental development, spiral development, rapid application development, extreme programming and agile methodology. Methodology used in this project is Systems Development Life Cycle (SDLC) which is Traditional methodology used to develop, maintain, and replace information systems. The activities or phases that will consider in SDLC methodology are: -

a. Planning
b. Analysis
c. Design
d. Implementation
e. Maintenance

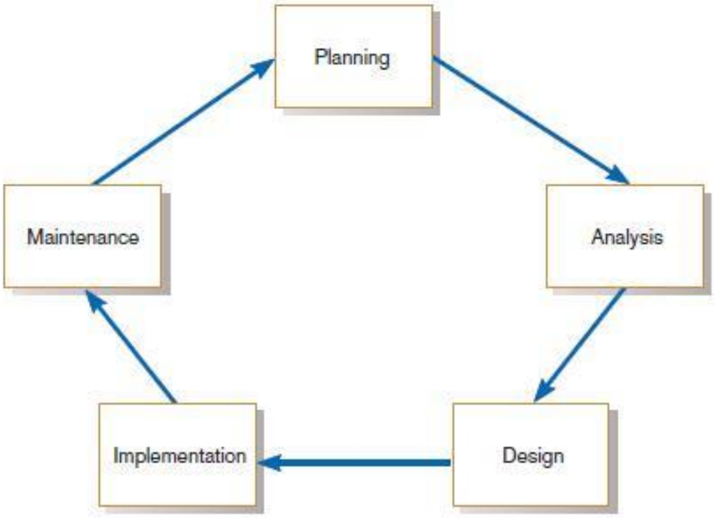


Figure 3.1: The systems development life cycle

3.2 TYPES METHODOLOGIES

3.2.1 Spiral Model Methodology

The spiral model is similar to the incremental model, with more emphasis placed on risk analysis. The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation. A software project repeatedly passes through these phases in iterations (called Spirals in this model). The baseline spiral, starting in the planning phase, requirements is gathered and risk is assessed. Each subsequent spiral builds on the baseline spiral.

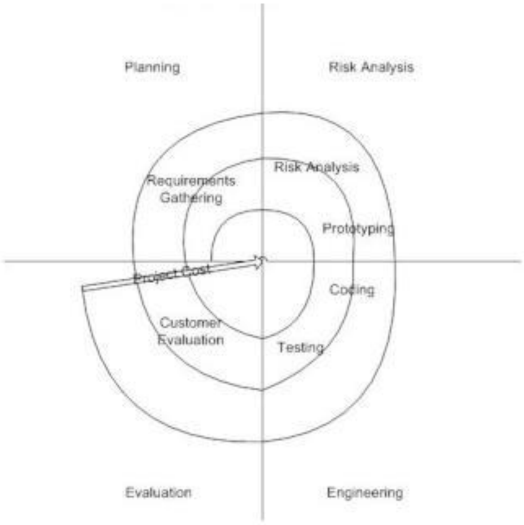


Figure 3.2: Spiral model

1. Planning Phase:

    - Requirements are gathered during the planning phase. Requirements like ‘BRS’ that is ‘Business Requirement Specifications’ and ‘SRS’ that is ‘System Requirement specifications’.

2. Risk Analysis:

    - In the risk analysis phase, a process is undertaken to identify risk and alternate solutions. A prototype is produced at the end of the risk analysis phase. If any risk is found during the risk analysis then alternate solutions are suggested and implemented.

3. Engineering Phase:

    - In this phase software is developed, along with testing at the end of the phase. Hence in this phase the development and testing is done.

4. Evaluation phase:

    - This phase allows the customer to evaluate the output of the project to date before the project continues to the next spiral.

The advantages of this methodology:

I. High amount of risk analysis hence, avoidance of Risk is enhanced.

II. Good for large and mission-critical projects.

III. Strong approval and documentation control.

IV. Additional Functionality can be added at a later date.

V. Software is produced early in the software life cycle.

Several disadvantages of this model:

I. Can be a costly model to use.

II. Risk analysis requires highly specific expertise.

III. Project's success is highly dependent on the risk analysis phase.

IV. Doesn't work well for smaller projects.

3.2.2 Waterfall Model Methodology

The waterfall model is a model, which was developed for software development; that is to create software. It is called as such because the model develops systematically from one phase to other in a downward fashion, like a waterfall.

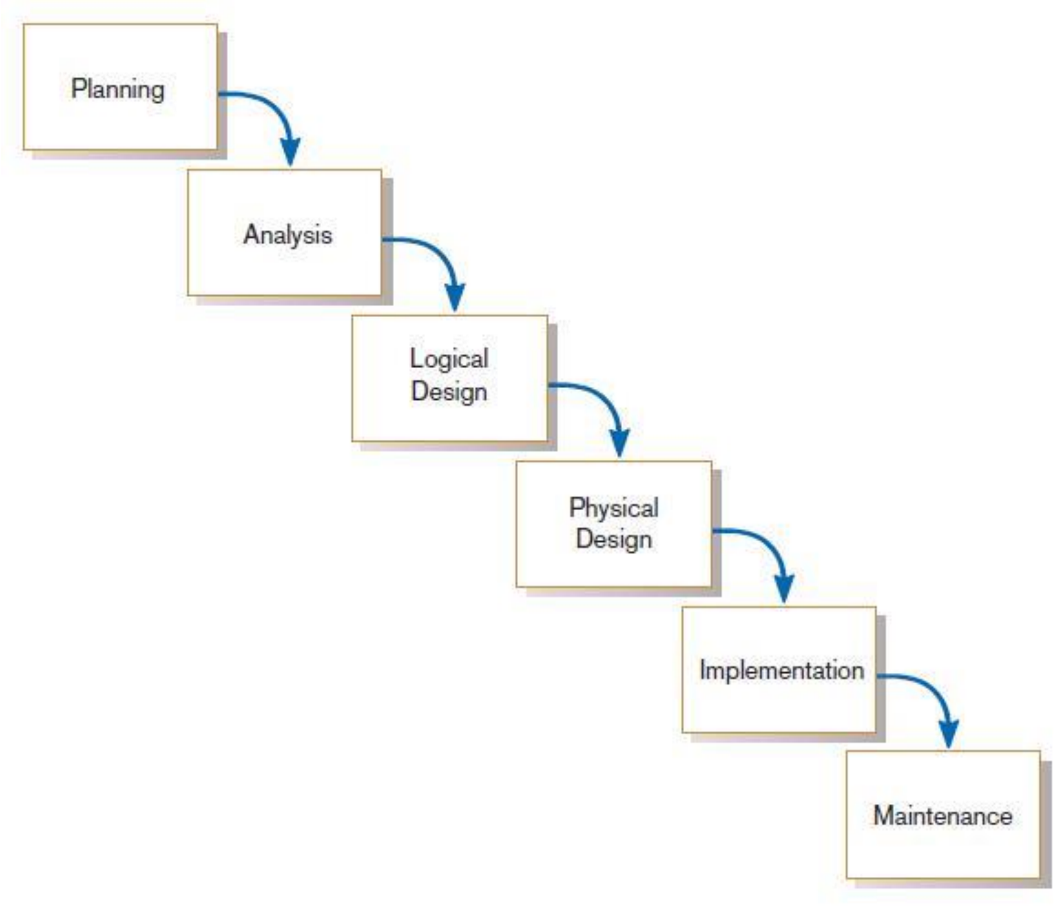


Figure 3.3: Waterfall model

The waterfall model has been organized on different eliminates particularly to help the product development organizations to build up a sorted out arrangement of development. By tailing this strategy, the venture will be isolated into numerous stages in this manner backing off the entire procedure.

The Details about each stage:

1. Planning:

    - In this stage, total information of any organization or system will be identified, analysed, prioritized and arranged carefully and thoroughly to ensure the outcome and result are parallel with the needed.

2. Analysis:

   - During this stage research is being conducted which includes brainstorming about the software, what it is going to be and what purpose is it going to fulfil.

3. Logical Design:

   - In this stage, all functional features of the system chosen for development in analysis stage are reviewed and begin the system design, identifying any additional system objects, determining operations and data structures for all objects, validating relationships and interactions between objects, and prototyping user interface objects.

4. Physical Design:

   - During this stage the logical specifications of the system from logical design are transformed into the technology-specific details from which all programming and system construction can be accomplished.

5. Implementation:

   - In this stage, the system will be coded, tested and installed. All of this will be documented for further use.

6. Maintenance:

   - During this stage, any misbehave or error in the system will be systematically repaired and improved based on the documentation provided in implementation stage.

The advantages of waterfall model methodology:

I. The project requires the fulfilment of one phase, before proceeding to the next. Therefore if there is a fault in this software it will be detected during one of the initial phases and will be sealed off for correction.

II. A lot of emphasis is laid on paperwork in this method as compared to the newer methods. When new workers enter the project, it is easier for them to carry on the work from where it had been left. The newer methods don’t document their developmental process, which makes it difficult for a newer member of the team to understand what step is going to follow next. The Waterfall Model is a straightforward method and lets one know easily what stage is in progress.

III. The Waterfall method is also well known amongst the software developers therefore it is easy to use. It is easier to develop various software through this method in short span of time.

Several disadvantages of this methodology:

I. Many software projects are dependent upon external factors; out of which the client for which the software is being designed is the biggest factor. It happens a lot of times, that the client changes the requirement of the project, thereby influencing an alteration in the normal plan of construction and hence the functionality too. The Waterfall Model does not work well in a situation like this as it assumes no alteration to occur once the process has started according to plan.

II. A huge amount of time is also wasted. For example if we study any software development process, we know that Phase II cannot be executed until Phase I has been successfully completed; so while the designers are still designing the software, time of the builders is completely wasted.

III. The testing period comes quite late in the developmental process; whereas in various other developmental programs the designs would be tested a lot sooner to find the flaw at a time when a lot of time and money has not been wasted.

IV. Elaborate documentation during the Waterfall method has its advantages, but it is not without the disadvantages as well. It takes a lot of effort and time, which is why it is not suitable for smaller projects.

3.2.3 Rapid Application Development (RAD)

RAD methodologies are methodology to radically decrease design and implementation time. In RAD it is involves extensive user involvement, prototyping, JAD sessions, integrated CASE tools, and code generators.

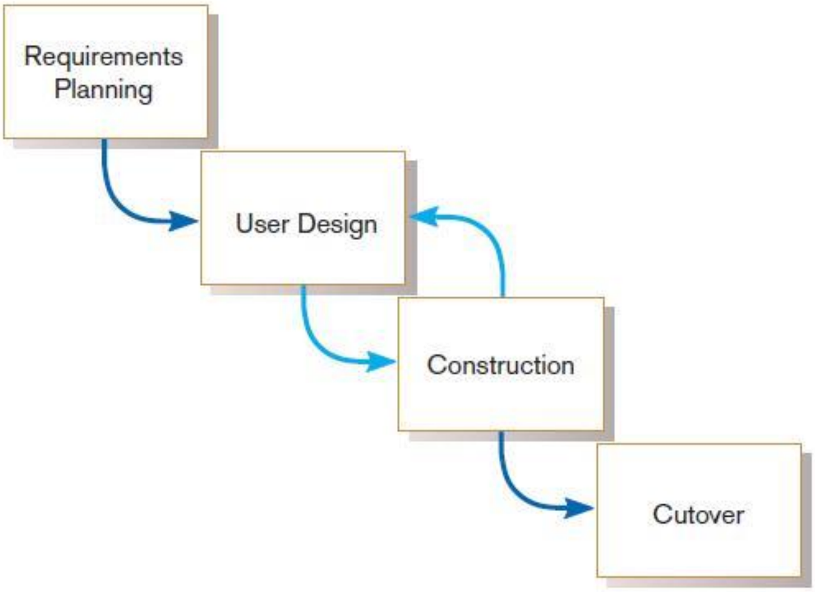


Figure 3.4: RAD life cycle

It is a type of incremental model. In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype.

1. Requirement Planning

The Plan stage forecasts the requirements for performing the project. Plan Project defines the project; produces the detailed project plan, project schedule, project organization, and resources. It identifies the project risks and ways to manage and mitigate them. It also obtains approval for the project and authorization to precede with the project activities. The approved project plan, including the budget and schedule, serves as a contract between the project manager and the project sponsor. This contract defines the scope of the project and what is required, the anticipated benefits, the resources required, and the performance-based rewards and penalties.

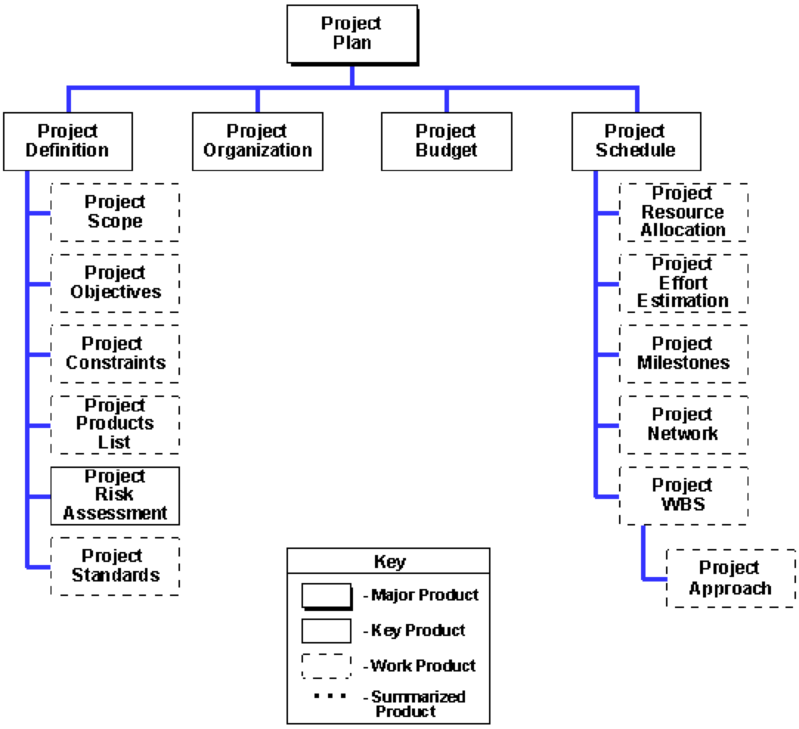


Figure 3.5: RAD Planning

2. User Design

The objectives of this stage are accomplished primarily by JAD workshops. In these workshops, the participants, who represent the end-users of the system and know the system area well, complete the detailed analysis of the business activities. These participants also develop the outline design of the system to the point where its construction and transition can be planned in detail. Finally, workshop participants review the implementation and transition plans.

The objectives of the User Design stage are to:

I. Analyse in detail business activities associated with the proposed system area

II. Analyse in detail the business data associated with the proposed system area

III. Develop the system structure in terms of the automated and manual procedures that will comprise the system

IV. Develop outline dialogs and proposed screen layouts for the most important automated system procedures

V. Select the appropriate construction approach for the system

VI. Prepare a work plan defining the steps necessary for transition of the system, the effort required to perform these steps, and a schedule by which these steps can be completed

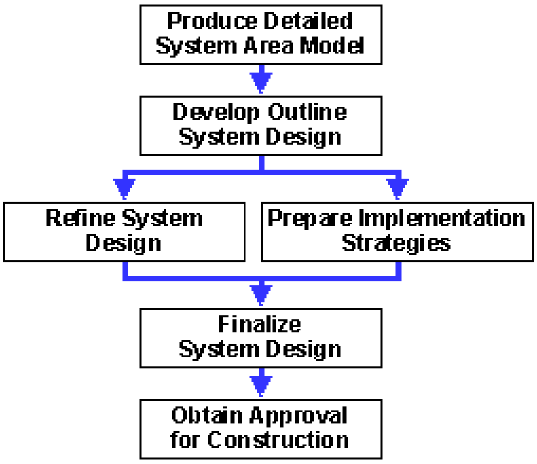

Produce Detailed System Area Model
Develop Outline System Design
Refine System Design
Prepare Implementation Strategies
Finalize System Design
Obtain Approval for Construction

Figure 3.6: RAD User Design

3. Construction

The design of the proposed system, initially described in the User Design stage, is completed in the Rapid Construction stage, and application software to implement that design is developed and tested. Activities to prepare for the transition of the system to production status are also performed.

The objectives of the Rapid Construction stage are to:

I. Complete a detailed design of the proposed system

II. Create and test the software that implements the proposed system

III. Generate a system that operates at an acceptable level of performance

IV. Prepare documentation necessary to operate the proposed application

V. Design, develop, and test the required transition software

VI. Perform the steps necessary to prepare for the conversion of the system to production status

Once the system is constructed and system tested, and the operating instructions and transition preparations are complete, acceptance tests are performed to confirm correct operation. Approval to proceed with the transition is then sought from the executive owner.

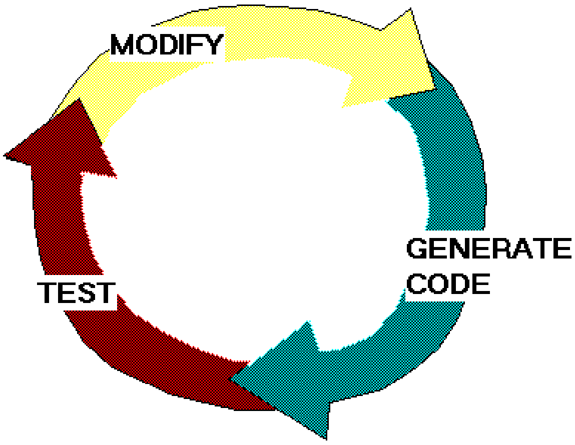


Figure 3.7: RAD Construction

The RAD developer needs to go through the fastest possible cycle of designing, generating, testing, modifying, and regenerating the system. The following diagram illustrates the sequence in which steps in the Rapid Construction stage must be conducted. Steps that may be conducted in parallel appear on the same horizontal row.

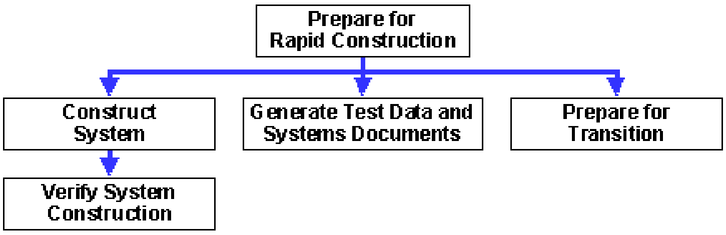

Prepare for Rapid Construction
Construct System
Generate Test Data and Systems Documents
Prepare for Transition
Verify System Construction

Figure 3.8: RAD Process flow

4. End Project / Cut Over

The End stage brings the project to an orderly conclusion and retains its history for the benefit of subsequent projects. End Project tasks archive the project materials, report on the project's performance, turn over the project results to the owners, and release the project resources for use on other projects. The following diagram illustrates the sequence in which steps in the Project Management End Project stage must be conducted. Steps that may be conducted in parallel appear on the same horizontal row.

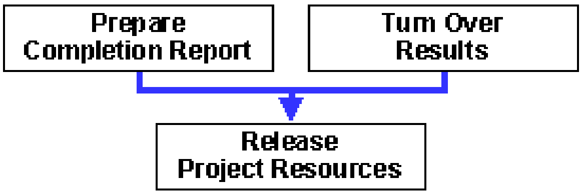


Figure 3.9: RAD End Project

The objectives of End Project are to:

I. Formally end the project in a controlled manner
II. Retain project history in the form of metrics, experiences, and lessons learned as input for future efforts
III. Gain acceptance of the project results

Advantages of the RAD model:

I. Reduced development time.
II. Increases reusability of components
III. Quick initial reviews occur
IV. Encourages feedback

Disadvantages of RAD model:

I. Depends on strong team and individual performances for identifying requirements.
II. Only system that can be modularized can be built using RAD
III. Requires highly skilled developers/designers.
IV. High dependency on modelling skills
V. Inapplicable to cheaper projects as cost of modelling and automated code generation is very high.

3.3 METHODOLOGY USED

From the three examples of SDLC process, this project uses RAD process this is because RAD suitable for this project when there is a need to create a system that can be modularized in 2-3 months of time such us this project do not have a long time period. Besides that, the next requirement to use this process is it can be used if there is high availability of designers for modelling and the budget is high enough to afford their cost along with the cost of automated code generating tools. Lastly, RAD SDLC model can with high business knowledge are available and there is a need to produce the system in a short span of time (2-3 months). Because of this requirement this project uses the RAD process.

A. Requirement Planning

In this project, requirement-planning starts with the project definition, which is consist of scope, objective, constraints, product list, risk assessment, and project standard. This stage is the critical stage to find all the information from user and also from client. The scopes of this project are done that stated at above. However, to find the scope of this project it needed to identify the benefit of this project to the user that will used this project and see the objective to develop this project. This project develops for the children that they used in learning aid for understanding about road safety.

Besides the project definition, the project schedule also important in this project development. This is because the project schedule consists of the resource allocation, effort estimation, Project milestones, network, and project WBS. This stage is important to identify the resource for this project. For example, resource headset for the virtual reality and information about the existing project. The effort estimation to identifying the number of people that's going to develop this project. However, this projects only a person doing from start to the end.

Because only a person that going thru this project the project organization is not very important because all the organization covered by the person. Lastly the project budget, this project only used own items so the budget of this project only for the small item and the small parts like paper, markers, and VR box with controller to place the smart phone to the VR box.

To get all the information from the user and also client, below the example of questionnaire that has been used to find the all information about this project and also the example question to the client that needs this project to develop. 30 respondents involved in this questionnaire which is from Sekolah Menengah Gambang from year 5 Fatonah.

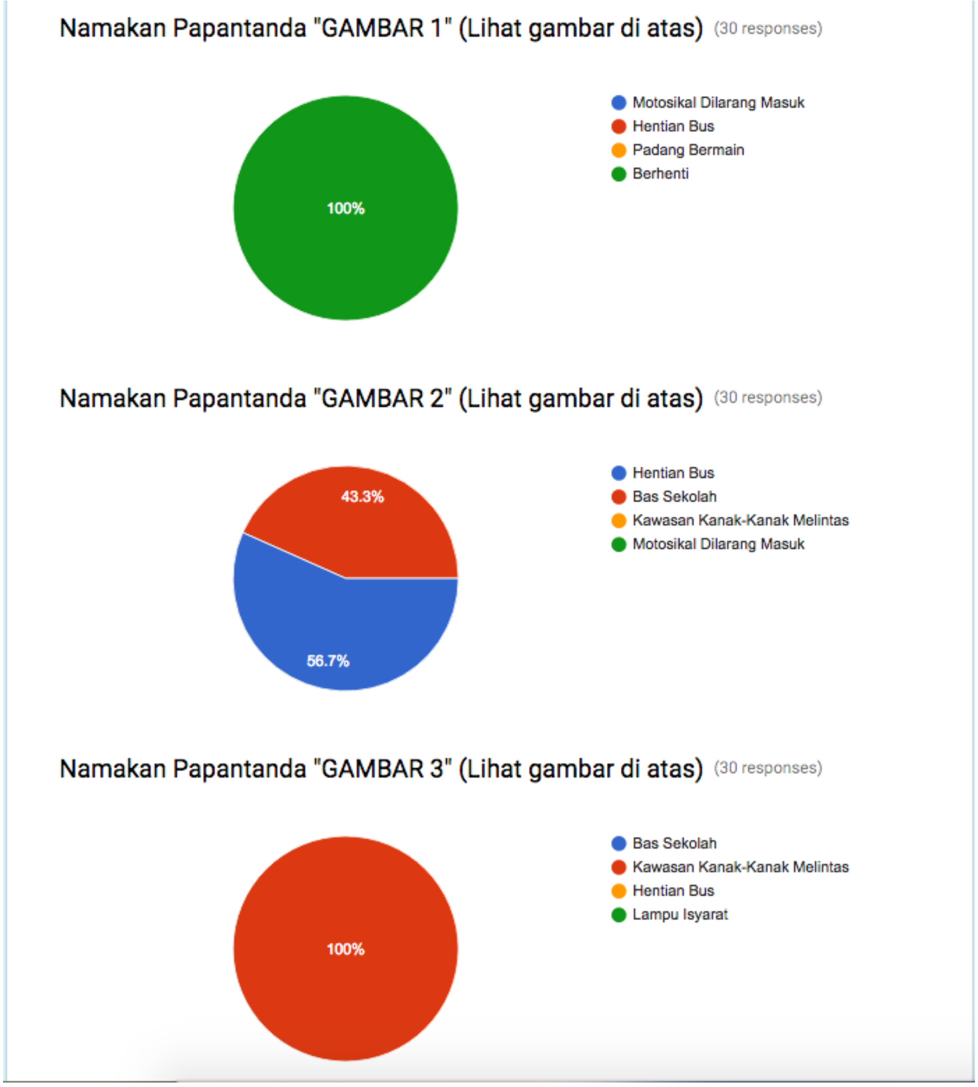
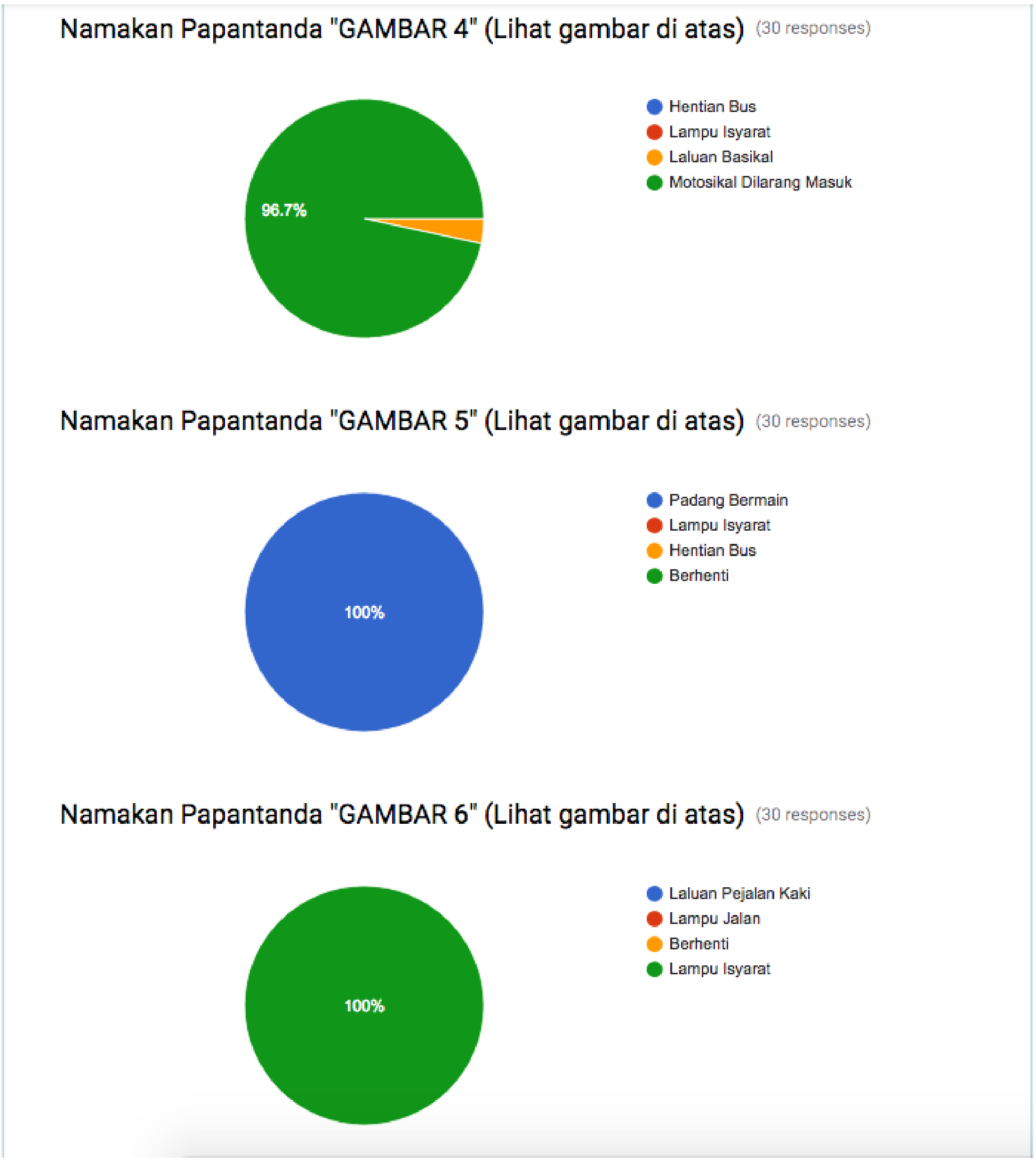


Figure 3.10: Questionnaire section one

From the first part of the questionnaire for the first question the result shows that 100% from all respondents have the correct answer. 43.3 % have a wrong answer and 56.7% has a correct answer for question two. Then the third question all the 30 respondent have the same and correct answer.



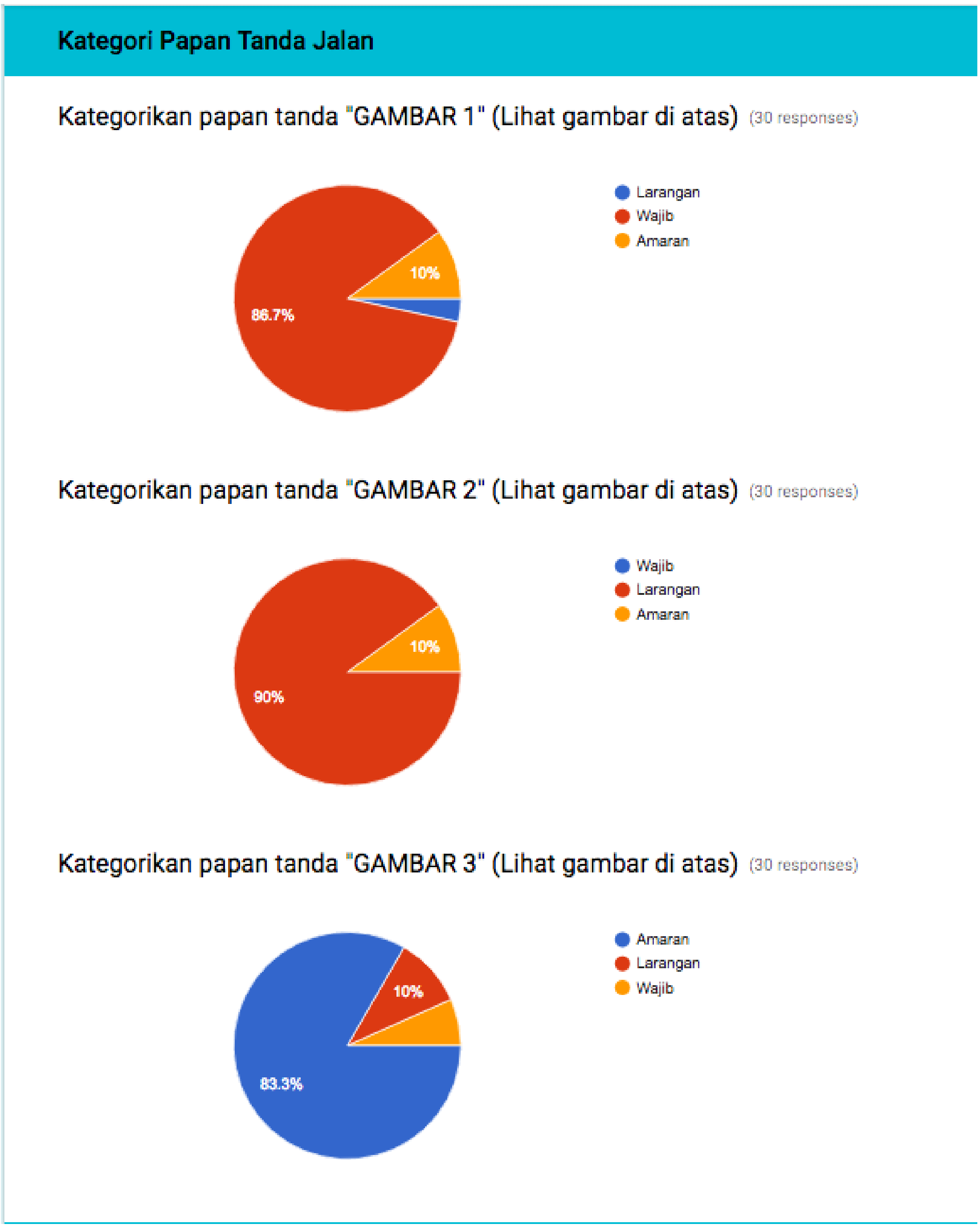
Namakan Papantanda "GAMBAR 4" (Lihat gambar di atas) (30 responses)

Namakan Papantanda "GAMBAR 5" (Lihat gambar di atas) (30 responses)

Namakan Papantanda "GAMBAR 6" (Lihat gambar di atas) (30 responses)

Figure 3.11: Questionnaire section one continue

For the forth question the result shows that 96.7% give the correct answer and the rest is wrong. Question five and six all the respondent give the correct answer. Can be concluding here, not all the students understand the meaning of the signboard that shows in the road.



Kategori Papan Tanda Jalan

Kategorikan papan tanda "GAMBAR 1" (Lihat gambar di atas) (30 responses)

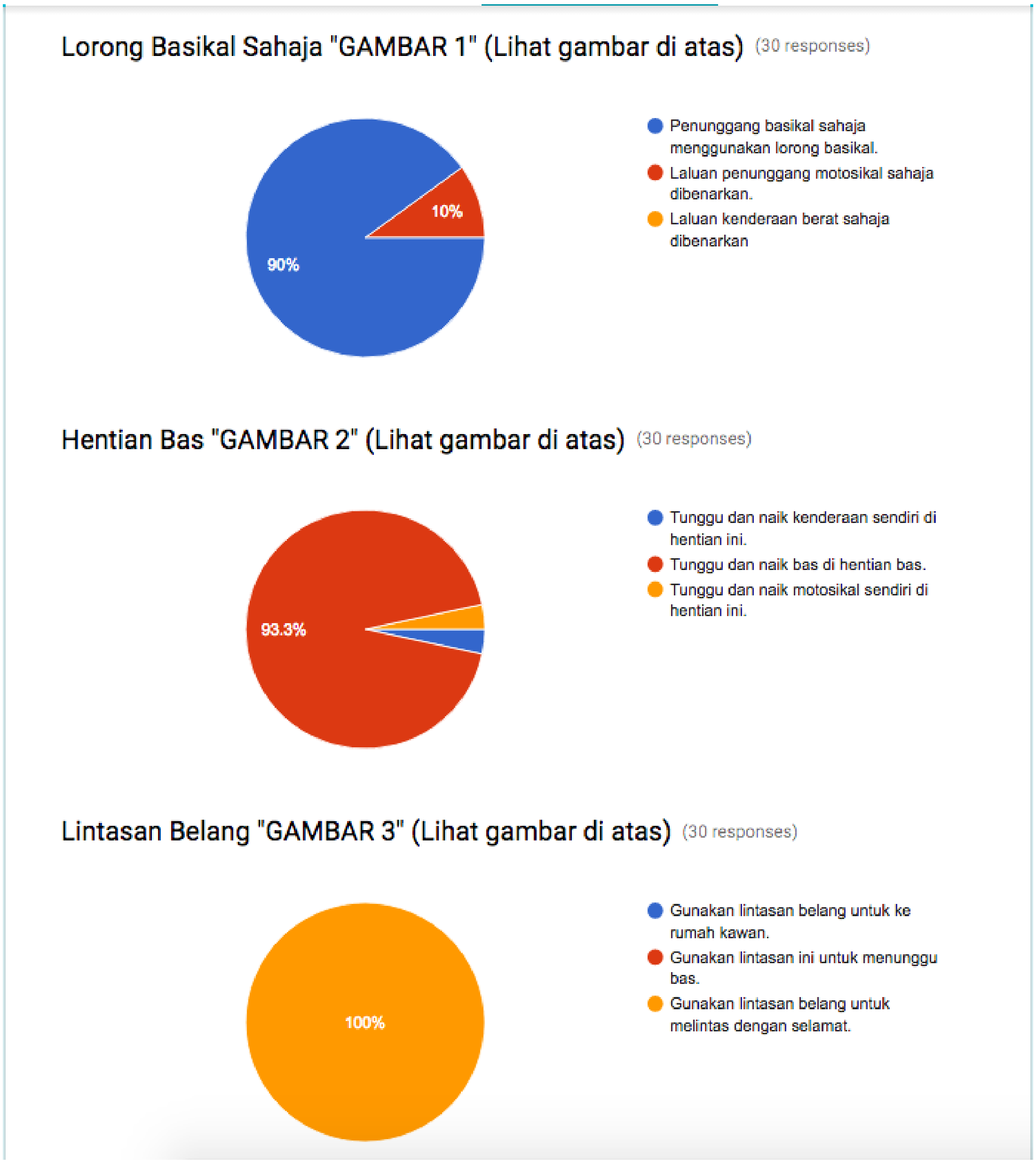
Kategorikan papan tanda "GAMBAR 2" (Lihat gambar di atas) (30 responses)

Kategorikan papan tanda "GAMBAR 3" (Lihat gambar di atas) (30 responses)

Figure 3.12: Questionnaire section two

For the second part of the questionnaire it is about the meaning of the colour by the signboard. For the first question only 86.7% that give the correct answer. Second question, 90% give the correct answer then 10% wrong answer. More than three over four give right answer then the rest wrong reply.

From this section of question, the result shows that more from the respondent do not know the meaning every colour of the signboards.



Lorong Basikal Sahaja "GAMBAR 1" (Lihat gambar di atas) (30 responses)

- ● Penunggang basikal sahaja menggunakan lorong basikal.
- ● Laluan penunggang motosikal sahaja dibenarkan.
- ● Laluan kenderaan berat sahaja dibenarkan

Hentian Bas "GAMBAR 2" (Lihat gambar di atas) (30 responses)

- ● Tunggu dan naik kenderaan sendiri di hentian ini.
- ● Tunggu dan naik bas di hentian bas.
- ● Tunggu dan naik motosikal sendiri di hentian ini.

Lintasan Belang "GAMBAR 3" (Lihat gambar di atas) (30 responses)

- ● Gunakan lintasan belang untuk ke rumah kawan.
- ● Gunakan lintasan ini untuk menunggu bas.
- ● Gunakan lintasan belang untuk melintas dengan selamat.

Figure 3.13: Questionnaire section three

The objective for the third part of the questionnaire is to measure the understanding and the right action can be takes from the signboards meaning. 90% from the respondent give the right answer and 10% give the wrong answer for the first question. The second question almost all respondent answer the right answer at 93.3%. The third question all respondent have a correct answer.

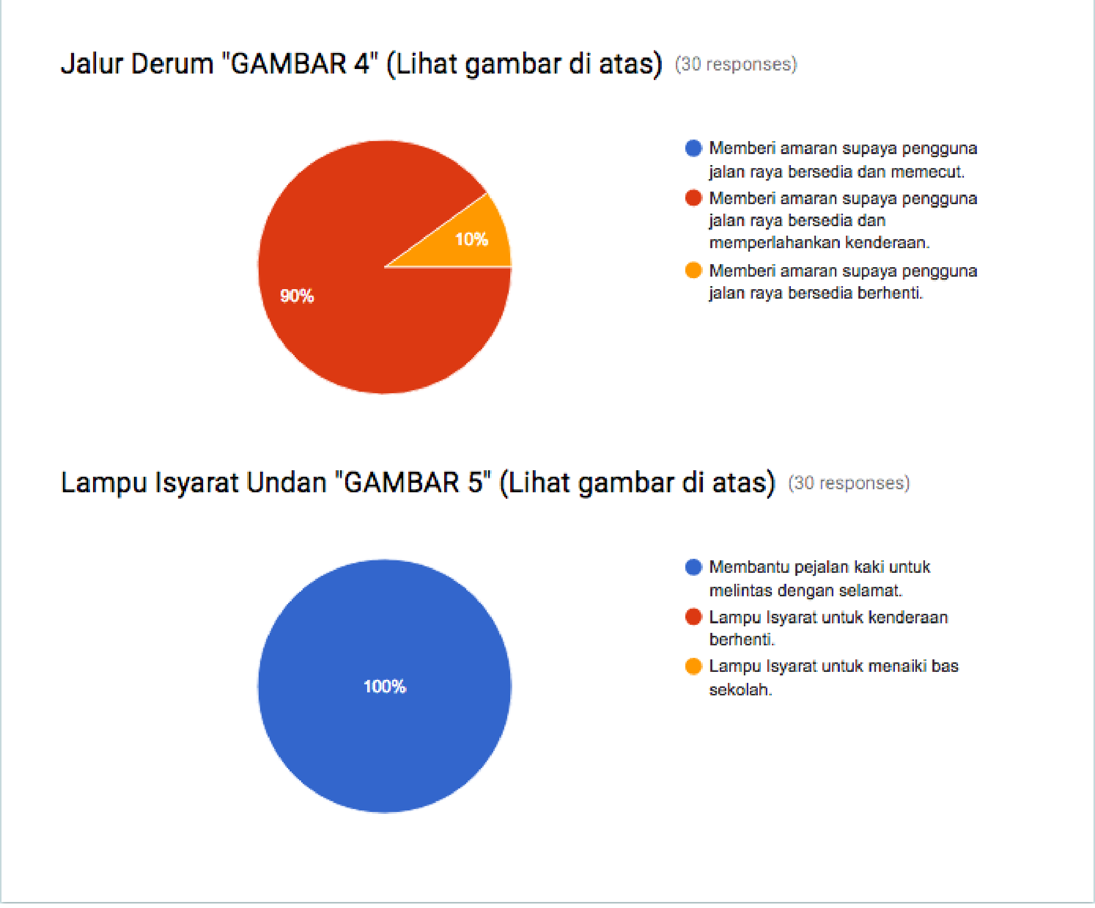


Figure 3.14: Questionnaire section three continue

For the forth question 90% respondent give the correct answer. Then the last question all respondent give the correct answer.

B. User Design

User design in the project is to analyse the details activity associated with the proposed area. Collect all the information from the client and children that want the system will look like to easier they used and more comfortable compare to the existing system. In this stage the structure of the system are develop to easy follow in construction stage. The draft design important to make sure the system of road safety fulfil all the needs from client and also the user that will use the system after finish develop. This is because, developing the system that not fulfil the client needed will make the system is useless.

The design of the system it will finalize the system design that will choose to continue to the construction stage. The criteria that make the system design will be chosen is the design have all the requirement system and the nearest. After finish, Prepare a work plan for this project to defining the steps necessary for transition of the project, the effort required to perform these steps, and a schedule by which these steps can be completed.

In this stage for this project shows the model and framework through the flow work. Figure 3.10 below shows the examples of planning design that will be create through this project. The VR box buys from the 11street.com and the smart phone use own smartphone that follow the minimum specification for VR application.

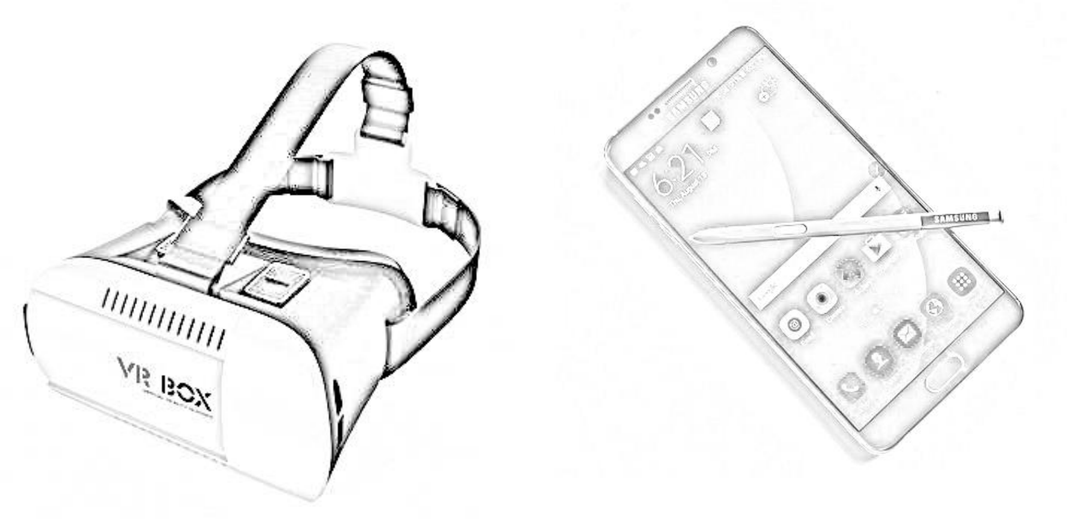


Figure 3.15: Example Design Position of the system

The diagram above shows the design of VR box and smartphone that will be install. The user will use the VR box and put the smartphone that has the application of VR and view from the VR box. The application will be run and the user view the application troughs the VR box and play the application by chooses the selection menu in the application. The user can immerse to the application by move their head to up – down, left and right at 360-degree view. Thus, the user can see the entire 3D object that has in the application. So, the users can follow the instruction and play with the application that gives knowledge about road safety.

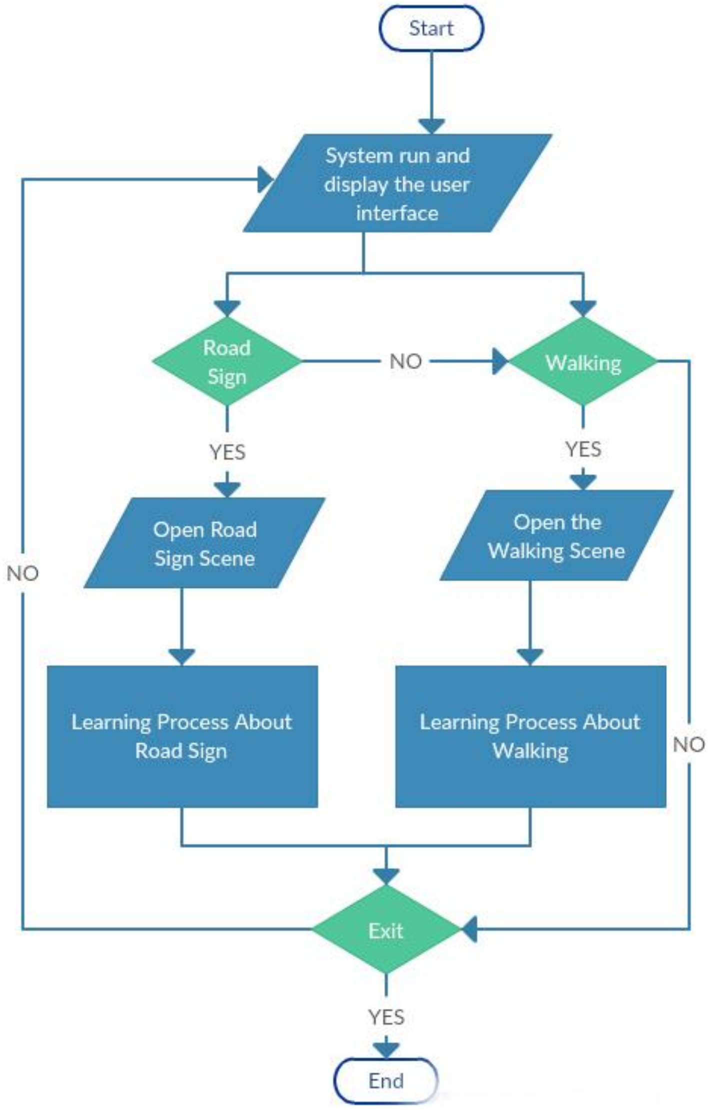

Start
System run and display the user interface
Road Sign
NO
Walking
YES
YES
Open Road Sign Scene
Open the Walking Scene
NO
Learning Process About Road Sign
Learning Process About Walking
NO
Exit
YES
End

Figure 3.16: Flow chart to run the application

Flow chart a diagram of the sequence of movements or actions of people or things involved in a complex system or activity. Thus, in this system the flow chart above show the how the system run the application to display the interface and show selection that user want to do in the application. In the first stage user must select between three conditions, which is road sign, walking, and cycling. This three different selection bring the user to the different scene in this application. Inside the scene the user will interact to the learning process that has in the application. The users will interact with the application by the controller that connect to the smartphone. So the user can learn about road safety in the different scene or different level or stage learning. After finish the user can jump again to the first selection menu to do another selection that have, if not the users can exit from the application by exit at the system. This flow chart shows how the system looks like and how the system will run in the smartphone and how the user can interact with it.

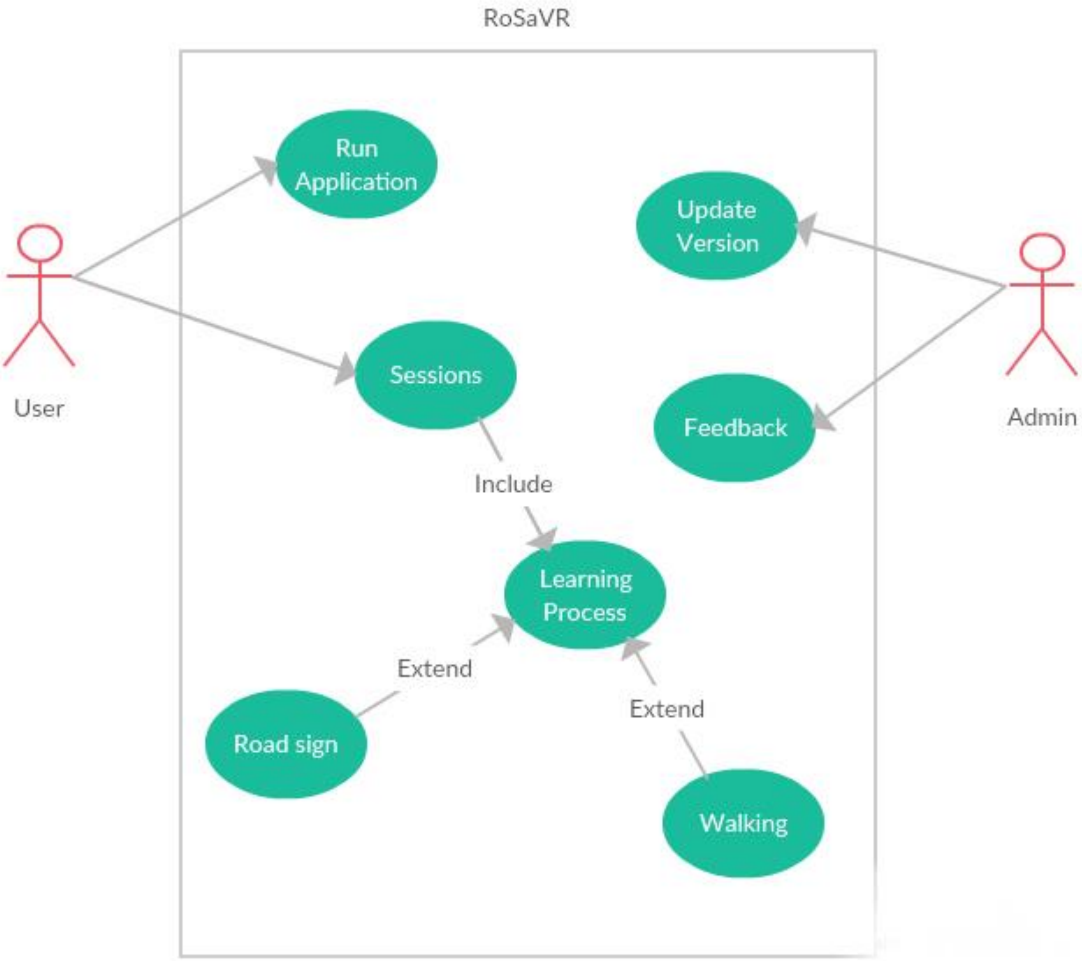


Figure 3.17: Use case diagram for the road safety application

The use case diagrams above show the application boundary and the use of the application. In this application only involve of two actors, which is user and also the admin. User in this application is the person that uses the application to learn about safety road that focus on kids firstly. The user can start the application to use it and inside the application user can choose which learning process that they want to lean first. After that, the application wills accesses the selection that has been done.

After accesses the scene, all the environment learning will shows to the user by full interactive view and user can interact with it by.

Besides, admin or developer also gives a big action in this application. However, the admin have a special function in this system is to manipulated the system and update the version for the application. Admin may use the feedback that given from user to make the new version or update the application by new version on it. This is because not all about road safety that included in this project. So, in the future admin can add more learning scene inside this application.

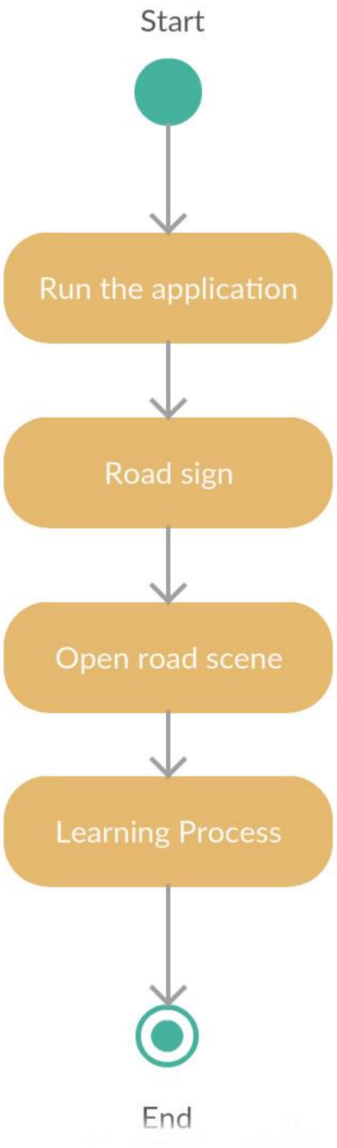
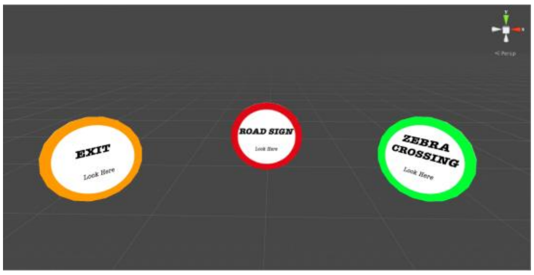
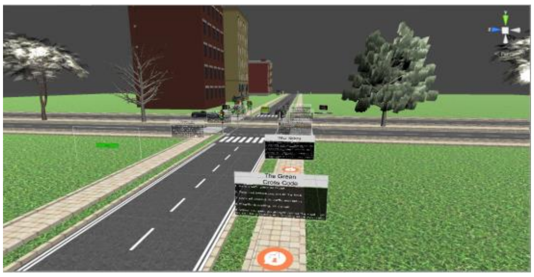


Figure 3.18: Activity Diagram for Road Sign Scene

Activity diagrams, which are related to program flow plans (flowcharts), are used to illustrate activities. In the external view, used of activity diagrams for the description of those business processes that describe the functionality of the business system. In this project the activity diagram shows how the system open the scene that related to user selected. Activity diagrams can be developed in various degrees of detail. However, the diagram above only show the activity diagram how the system show the scene and show the learning process that consist of 3D object road sign and the instruction from the application. The diagram above shows captures the dynamic behaviour of the system.

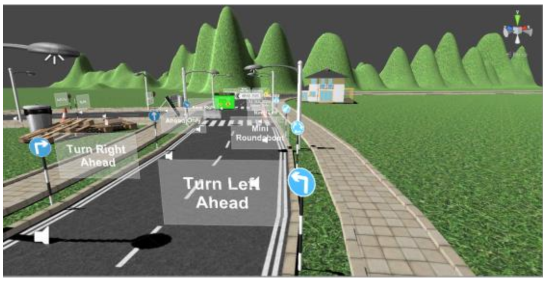
Storyboard 1: Ready Application APPS WIDGETS Communication Google Music & Audio Photography Productivity Tools Themes Photos RoSaVR Street View	**Graphic**: - **Animation**: - **Audio**: - **Interactivity**: -
Description: The application already installs inside the smartphone and click it to run the application.	

Storyboard 2: Interface  EXIT Look Here ROAD SIGN Look Here ZEBRA CROSSING Look Here	**Graphic**: Background Image **Animation**: - **Audio**: Background music **Interactivity**: Three button selection one button exit
Description: Users need to choose the selection to continue the learning, the three selections is, road sign, walking and cycling.	

Storyboard 3: Walking 	**Graphic**: Background Image / Town **Animation**: Car Moving on the road **Audio**: Background music / car engine start and moving **Interactivity**: Walking around the road
Description: User walking around the road with the true direction. User must walking at the walking line area and must opposite direction with the car and vehicle.	

<table>
<tr><td><b>Storyboard 4: Walking Direction</b>
 </td><td><b>Graphic</b>: Background Image / Town
<b>Animation</b>: Car Moving on the road
<b>Audio</b>: Background music / car engine start and moving
<b>Interactivity</b>: Walking around the road and cross road</td></tr>
<tr><td colspan="2"><b>Description</b>:
User walking around the road with the true direction. User must walking at the walking line area and must opposite direction with the car and vehicle.</td></tr>
</table>

<table>
<tr><td><b>Storyboard 5: Zebra Crossing</b>
 </td><td><b>Graphic</b>: Background Image / Town
<b>Animation</b>: Car Moving on the road
<b>Audio</b>: Background music / car engine start and moving
<b>Interactivity</b>: Walking around the road and cross road</td></tr>
<tr><td colspan="2"><b>Description</b>:
User walking around the road with the true direction. User must walking at the walking line area and must opposite direction with the car and vehicle. Apply the step how to cross the road without zebra crossing.</td></tr>
</table>

<table>
<tr><td><b>Storyboard 6: Road Sign</b>
 </td><td><b>Graphic</b>: Background Image and Road sign
<b>Animation</b>: Animation fade in and fade out of the road sign
<b>Audio</b>: Background music
<b>Interactivity</b>: Next and Previous button exit button, and back button.</td></tr>
</table>

Description: User controls the cycling around the road with the true direction. User must cycle at the cycling line area. Apply the rules and step how to cycling at the road that has many car and vehicle.

<table>
<tr><td><b>Storyboard 7: Traffic light</b>
 </td><td><b>Graphic</b>: Background Image / traffic light
<b>Animation</b>: Changing colour of traffic light
<b>Audio</b>: Background music
<b>Interactivity</b>: -</td></tr>
<tr><td colspan="2"><b>Description</b>:
The colour of traffic light will change with the description too. Give understand about the important traffic light and what the colour mean to every each colour.</td></tr>
</table>

<table>
<tr><td><b>Storyboard 9: Traffic light for Pedestrian</b>
 </td><td><b>Graphic</b>: Background Image / traffic light
<b>Animation</b>: Changing colour of traffic light
<b>Audio</b>: Background music
<b>Interactivity</b>: -</td></tr>
<tr><td colspan="2"><b>Description</b>:
The colour of traffic light will change with the description too. Give understand about the important traffic light and what the colour mean to every each colour. Focus to the pedestrian traffic light.</td></tr>
</table>

C. Construction

The construction stage in this project is the development system that the design stage proposed. In this stage all the code that will use in VR will be tasted. The entire 3D object will imbed with the code and also the VR application that put into the smartphone. In this project the flow or transition software that will use to test the system is generate code, test the code and modify the code if the code do not work or the code to not work like what the designer system needed. This stage also will prepare documentation necessary to operate the proposed system and perform the steps necessary to prepare for the conversion of the system to production status.

D. End Project / Cut Over

In the cut over stage, all the complete report about this project must be prepare and the turn over results also must be prepared.

3.4 HARDWARE AND SOFTWARE

Software approach is a description about the software used during the development process of this project. The software that are needed to develop this project is Blander and Autodesk Maya to create a 3D modelling of objects, Unity3D; used as a platform to create a virtual environment, Google cardboard SDK to create camera that has stereoscopic view to generate the different view from different eyes.

3.4.1 Hardware requirement

1. Computer

Operating Systems

- Windows 7 or 8 / IMAC

Minimal specs for Hardware

- 32 bits, Dual Core CPU with at least 2 GHZ, and SSE2 support.
- 2 GB RAM
- 24 bits 1280x768 display
- Mouse or Track pad
- OpenGL Graphics Card with 256 MB RAM

Average specs for Hardware

- 64 bits, Quad Core CPU
- 8 GB RAM
- Full HD Display with 24 bit colour
- Mouse or Track pad
- OpenGL Graphics Card with 1 GB RAM

Production specs for Hardware

- 64 bits, Dual 8 Core CPU
- 16 GB RAM
- Two times Full HD Display with 24 bit colour
- Mouse + tablet
- Dual OpenGL Graphics Cards, quality brand with 3 GB RAM

2. VR box and Controller

Figure 3.19: VR box body

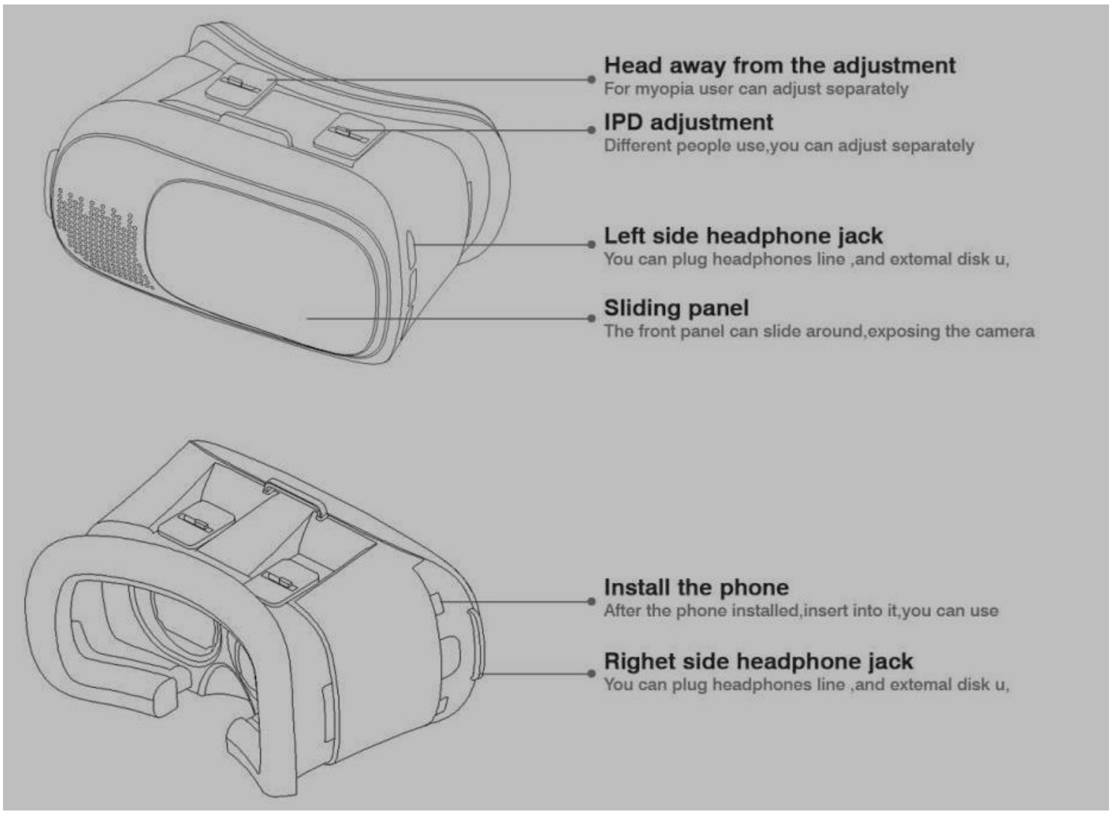

Head away from the adjustment
For myopia user can adjust separately
IPD adjustment
Different people use,you can adjust separately
Left side headphone jack
You can plug headphones line ,and extemal disk u,
Sliding panel
The front panel can slide around,exposing the camera
Install the phone
After the phone installed,insert into it,you can use
Righet side headphone jack
You can plug headphones line ,and extemal disk u,

Figure 3.20: VR Box architectures

Figure 3.21: VR box Controller

3. Smartphone

Display

Screen size (inches)	5.70
Touchscreen	Yes
Resolution	1440x2560 pixels
Pixels per inch (PPI)	518

Hardware

Processor	5GHz octa-core
Processor make	Exynos 7420
RAM	4GB

Software

Operating System	Android 5.1.1

Sensors

Proximity sensor	Yes
Accelerometer	Yes
Ambient light sensor	Yes
Gyroscope	Yes
Barometer	Yes

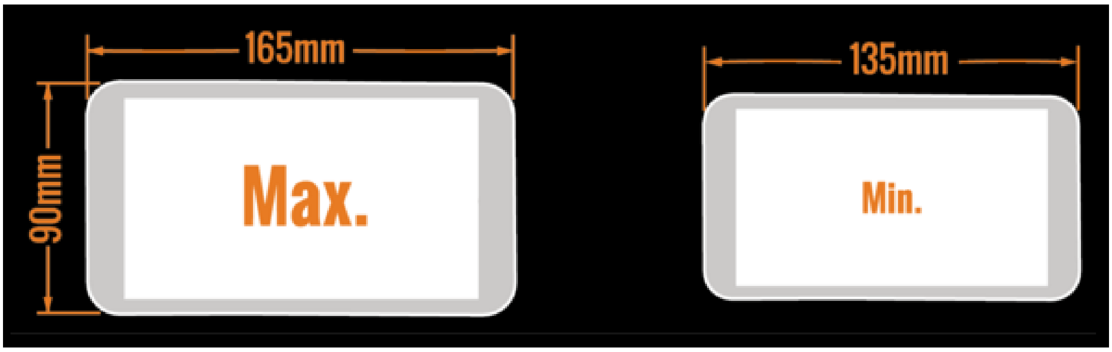


Figure 3.22: Size of smartphone that can fit in into the VR box

3.4.2 Software requirement

1. Blender

Blender is a professional free and open-source 3D computer graphics software product used for creating animated films, visual effects, art, 3D printed models, interactive 3D applications and video games. Blender's features include 3D modelling, UV unwrapping, texturing, raster graphics editing, rigging and skinning, fluid and smoke simulation, particle simulation, soft body simulation, sculpting, animating, match moving, camera tracking, rendering, video editing and compositing. Alongside the modelling features it also has an integrated game engine.

To get real movement in prayers, blender3D animation technique is used. In technique animation, movement per unit calculated through key frame namely equivalent to per second real-time. Here every movement needs to be detailed so that the movement can be same like real life people praying.

2. Unity

   Unity is a cross-platform game creation system developed by Unity Technologies, including a game engine and integrated development environment (IDE). It is used to develop video games for web sites, desktop platforms, consoles, and mobile devices. Unity is notable for its ability to target games to multiple platforms.

3. Java Programming Language and C#

   Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable.

4. Google Cardboard SDK

   The Google cardboard Software Development Kit (SDK) allows the developer to use Unity to create virtual reality experiences for Android and iOS apps running in Google Cardboard. The SDK has plugin scripts and prefabs that make easy to:

I. Begin a new VR Unity project from scratch
II. Adapt an existing Unity 3D application to VR
III. Make an app that can easily switch in and out of VR mode

The integration with the Cardboard SDK for Android provides:

I. User head tracking
II. Side-by-side stereo rendering
III. Detecting Cardboard-only user inputs such as the trigger
IV. Automatic stereo configuration for a specific Cardboard model
V. Distortion correction for Cardboard lenses
VI. An alignment marker to help centre the screen under the lenses

VII. A settings button that links to the Cardboard app for managing headset parameters

VIII. Automatic gyro drift correction

The plugin provides the following features:

I. Simulate a user's head movement in Unity play mode using the mouse and alt/control keys to pan or tilt the camera.

II. Control the approximate field of view regardless of differences in Cardboard device configurations.

III. Respect camera zoom effects in a VR-safe manner, that is, compatible with head tracking.

IV. Properly support picture-in-picture windows in stereo, and adjust their screen position so they remain visible after distortion correction.

V. Dynamically adjust the stereo level to reduce eyestrain.

VI. Easily determine the direction the user is gazing, for example, to cast rays for hit detection.

VII. Interaction with Canvas UI elements using gaze and the trigger.

VIII. Use a single code path for both trigger and screen tap detection.

IX. Support for Image Effects and Deferred Rendering, or bypassing this support for increased FPS.

5. Photoshop

Adobe Photoshop is an image-editing program used to perform a variety of functions in the graphics, photography and digital art worlds. It was released by Adobe Systems, Inc. and marketed as the new standard for personal image editing and paint programs. In this project the Photoshop used when to develop the banner of the project before presentation day.

6. Autodesk Maya

   Maya 3D animation, modelling, simulation, rendering, and compositing software offers a comprehensive creative feature set on a highly extensible production platform. Maya provides high-end character and effects toolsets along with increased productivity for modelling, texturing, and shading creation tasks.

3.5 GANTT CHART

This Gantt Chart Show research phases from starting till project completion. This project Gantt chart involves two phases, which is first phase about documentation and second phase about implementation. The green colour is Gant Chart PSM I and the purple is PSM II.

3.6 SUMMARY

Basically, this project will help in education and become one of the learning aids and help kids to know about the road safety. This project also, implements the technology that has to day into the learning process and combine between to learning process which is book and virtual reality technology. This project also will give benefit because it tried to keep the understanding about road safety in educations now a day. However, this project also has a limitation and only focusing in small part in road safety. It is only cover the several topics in road safety and focusing to give understanding for the kids.

This project also study about the existing system that already implement virtual reality and also the existing that help in education about road safety. This project also covers the hardware and software that will be used in development this project after this. All the software and hardware already stated. Because of the implementation time is short, so the methodology that used in this project is rapid application development (RAD). Lastly, hope this project will help and will be implemented to the target user and help in education field to make road safety knowledge more easy to learn and more easers understand compare to the existing system that has today.

CHAPTER 4

IMPLEMENTATION

4.1 INTRODUCTION

In this chapter will discuss about the process of development this project. All the process will be discussed from nothing until the project complete develops. This project consists of three levels or three scenes and one interface scene to be developing and every scene has their own purpose but it connected to each other. All the implementation process only use the software and hardware that already stated at literature review scopes. To make the development process outcome same like what this project propose it must follow the storyboard and the follow the Gantt chart to finish it on time.

4.2 IMPORTING SDK DEVELOPMENT

In this stage it wills discuss how to create the interface of this project. For the stating just double click to the Unity icon after installed to the computer because this software is free but it is need our email to download it. After register the email start to develop new project and rename base on the suitable name for this project it call RoSaVR. After creating project automatically the Unity create new scene for the user. Then user can start develop their project.

However, in tis case, this project needs SDK (Software Development Kit) before start it. In this development the important stage is downloads the cardboard SDK for Unity from the Google developer site. After competed downloading the SDK, import the SDK into the Unity. Below is the image of Unity interface when it started open. To import the SDK go to asset on the menu bar Unity and select the import package from the drop down menu then select custom packaged and find the SDK that already download that located in the computer.

(**) If the SDK already imported into Unity the "cardboard" folder will be appear in the "project" window. In the "project" window click the cardboard folder then click the prefabs then choose "Cardboard Main" then drag into the Unity scene. Delete the "Main Camera" in the "Hierarchy" windows; just leave the cardboard main and the Directional light.

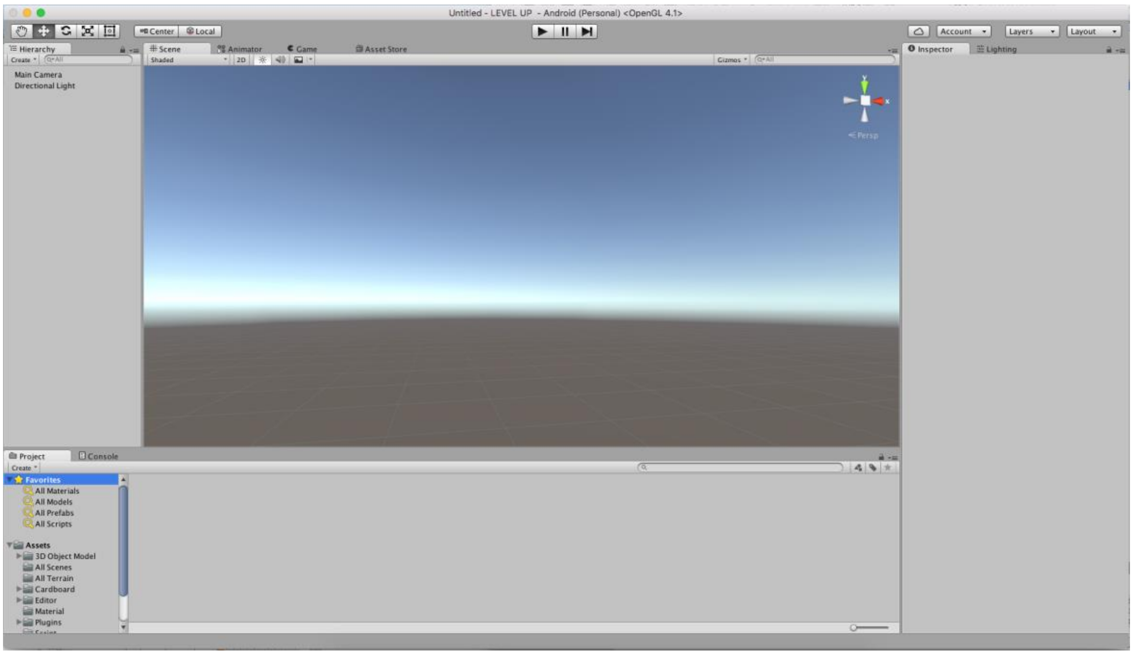


Figure 4.1: Unity interface

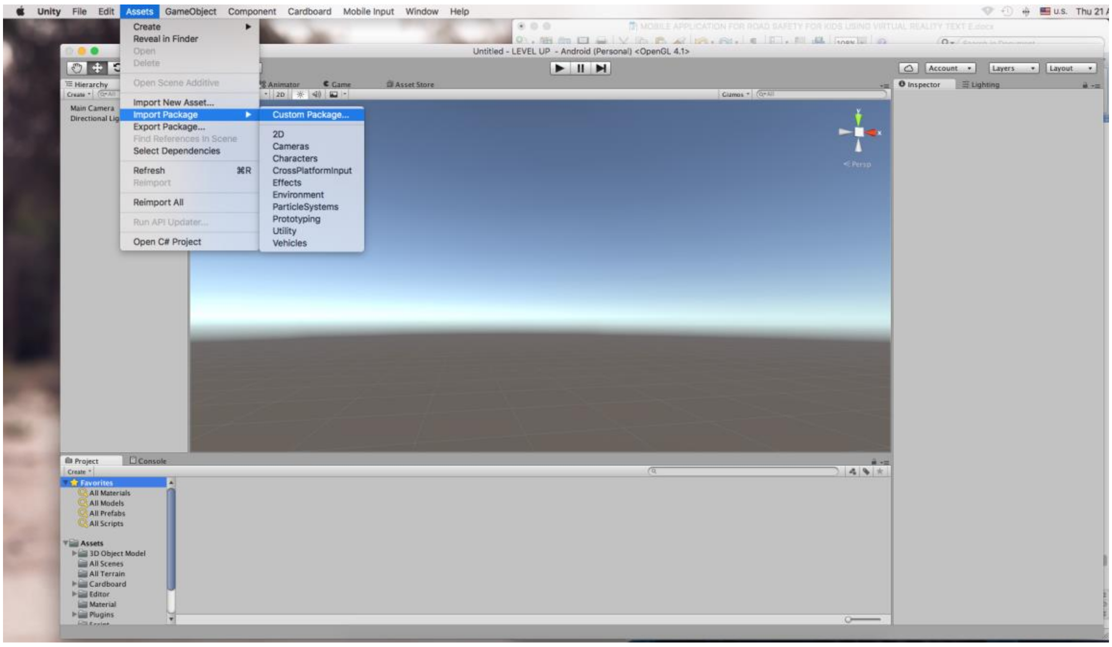


Figure 4.2: How to import SDK into Unity

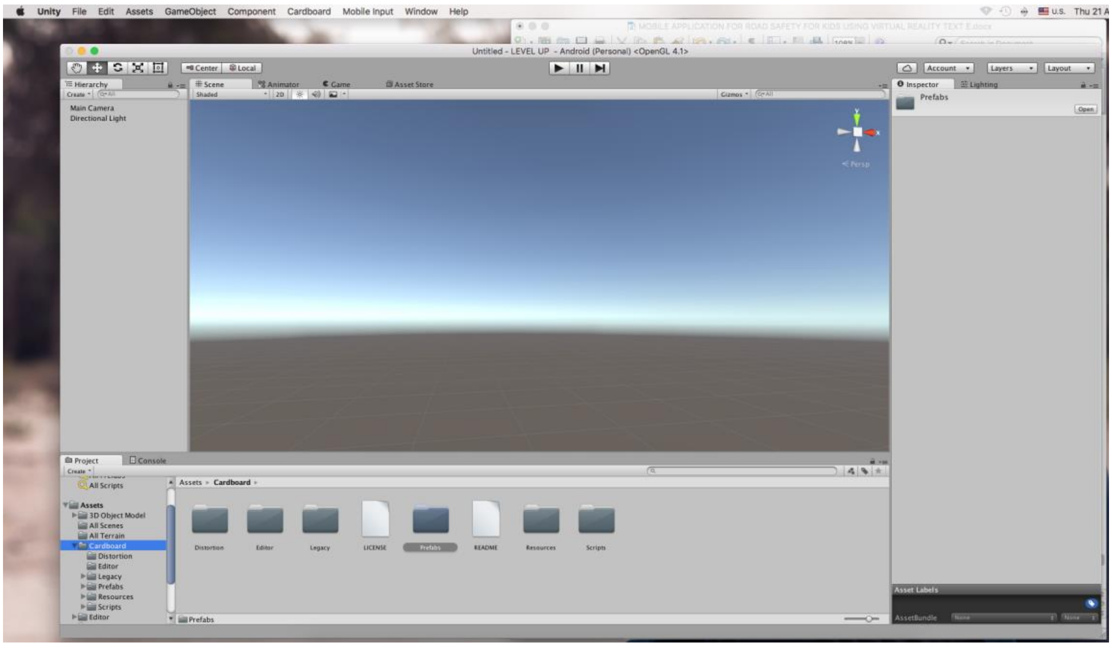
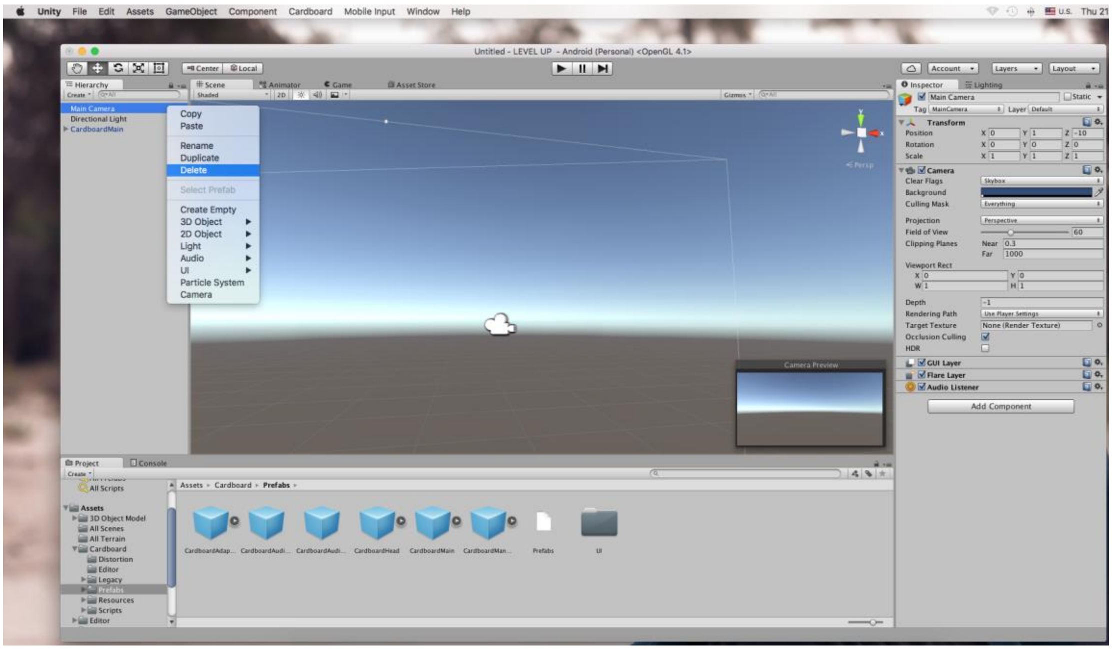


Figure 4.3: Folder cardboard in Project Windows



☑ Main Camera □ Static
Tag MainCamera Layer Default
Transform
Position X 0 Y 1 Z -10
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
☑ Camera
Clear Flags Skybox
Background
Culling Mask Everything
Projection Perspective
Field of View 60
Clipping Planes Near 0.3 Far 1000
Viewport Rect X 0 Y 0 W 1 H 1
Depth -1
Rendering Path Use Player Settings
Target Texture None (Render Texture)
Occlusion Culling ☑
HDR □
☑ GUI Layer
☑ Flare Layer
☑ Audio Listener

Figure 4.4: Drag in the Cardboard Main into scene and delete the main camera.

After finished following the steps to import SDK hit the play button on the top of Unity window and hit it again to stop the play game scene. After hit the play button the game scene will open that the view from our cardboard Main camera and it divided into two parts which is left eye view and right eye view.

4.3 INTERFACE DEVELOPMENT

Before starting the development, follow the 4.1 stapes above the import the SDK into Unity.

After finished follow the steps, start with create the empty game object at the “Hierarchy” windows by right click the windows and select the “Create Empty”. Click the Create empty and rename it to “StartGameObject”. After rename it the empty game object will appear in the scene (scene in the Unity the place that user can manipulate their object and do other things). For example, the object position, location, rotation and the size of the object can be change by click the object and use "W" for change location, "E" for change the rotation and "R" for change the size of the object. Then use "Q" for change viewing to editing and use "F" for focus the object to more near to the manipulation tools. If need to change the location, position, rotation and size by using manual it can be done at the inspector window after clicking the object that need to be manipulate. Then create another one and rename it to "ExitGameObject".

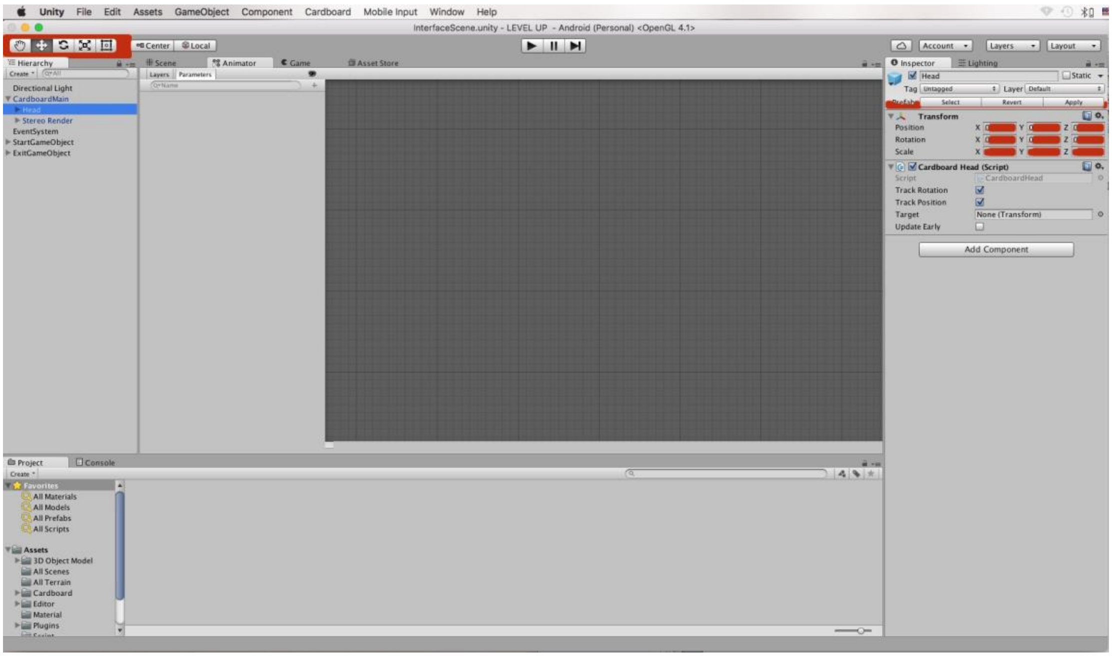


Figure 4.5: Manipulation tool in Unity

Now there have two empty game objects. Now right click at "StartGameObject" select the UI from the menu that has and select canvas and rename it "StartMenu". Right click the "StartMenu" then choose UI and select button then rename it into "RoadSignButton". Repeat it to make four buttons and rename it based on the scene name. Every each button will make the user go to the different scene that has in this project. To the same things to the "ExitGameObject" create canvas then rename it to "QuitMenu" then create two buttons inside the "QuitMenu" canvas. And rename the buttons "YesButton" and "NoButton". In the button inspector windows at the right top Unity window the colour button and the type of font also the text in the button can be change. The inspector main function is like the main setting of the button. Every item that has created in the unity scene has their own inspector and they are on setting.

Every button that has created must be having the even system to take the user input by clicking on that button. To do that, the coding must be creating to make sure the button is functioning. Below is the example of coding that used in this project.

```
using UnityEngine;
using System.Collections;
using UnityEngine.SceneManagement;

public class RedialCutoutMenuExit : MonoBehaviour {

    // How long to look at Menu Item before taking action
    public float timerDuration = 2f;

    // This value will count down from the duration
    private float lookTimer = 0f;

    // My renderer so I can set _Cutoff value
    private Renderer myRenderer;

    // Box Collider
    private BoxCollider myCollider;

    // Is player looking at me?
    private bool isLookedAt = false;

    // MonoBehaviour Start
    void Start() {
        // My Collider
        myCollider = GetComponent<BoxCollider>();
        // Get my Renderer
        myRenderer = GetComponent<Renderer>();
        // Set cutoff
        myRenderer.material.SetFloat("_Cutoff", 0f);
    }

    // MonoBehaviour Update
    void Update() {
        // While player is looking at me
        if (isLookedAt) {
            // Reduce Timer
            lookTimer += Time.deltaTime;

            // Set cutoff value on material to value between 0 and 1
            myRenderer.material.SetFloat("_Cutoff", lookTimer / timerDuration);

            if (lookTimer > timerDuration) {
                // Reset timer
                lookTimer = 0f;

                // disable collider
                myCollider.enabled = false;

                // Do something
                Debug.Log("BUTTON HAS BEEN SELECTED!");
                Application.Quit ();
                // Disappear
                gameObject.SetActive(false);
            }
        }  else {
            // Reset Timer
            lookTimer = 0f;
```

```
            // Reset Cutoff
            myRenderer.material.SetFloat("_Cutoff", 0f);
        }
    }

    // Google Cardboard Gaze
    public void SetGazedAt(bool gazedAt) {
        // Set the local bool to the one passed from Event Trigger
        isLookedAt = gazedAt;
    }
}


.............................................................................................................
..

using UnityEngine;
using System.Collections;
using UnityEngine.SceneManagement;

public class RedialCutoutMenuRoad : MonoBehaviour {

    // How long to look at Menu Item before taking action
    public float timerDuration = 2f;

    // This value will count down from the duration
    private float lookTimer = 0f;

    // My renderer so I can set _Cutoff value
    private Renderer myRenderer;

    // Box Collider
    private BoxCollider myCollider;
```

```
// Is player looking at me?
private bool isLookedAt = false;

// MonoBehaviour Start
void Start() {
    // My Collider
    myCollider = GetComponent<BoxCollider>();
    // Get my Renderer
    myRenderer = GetComponent<Renderer>();
    // Set cutoff
    myRenderer.material.SetFloat("_Cutoff", 0f);
}

// MonoBehaviour Update
void Update() {
    // While player is looking at me
    if (isLookedAt) {
        // Reduce Timer
        lookTimer += Time.deltaTime;

        // Set cutoff value on material to value between 0 and 1
        myRenderer.material.SetFloat("_Cutoff", lookTimer / timerDuration);

        if (lookTimer > timerDuration) {
            // Reset timer
            lookTimer = 0f;

            // disable collider
            myCollider.enabled = false;

            // Do something
```

```
                Debug.Log("BUTTON HAS BEEN SELECTED!");
                SceneManager.LoadScene (1);
                // Disappear
                gameObject.SetActive(false);
            }
        }  else {
            // Reset Timer
            lookTimer = 0f;
            // Reset Cutoff
            myRenderer.material.SetFloat("_Cutoff", 0f);
        }
    }

    // Google Cardboard Gaze
    public void SetGazedAt(bool gazedAt) {
        // Set the local bool to the one passed from Event Trigger
        isLookedAt = gazedAt;
    }
}
```

....
..

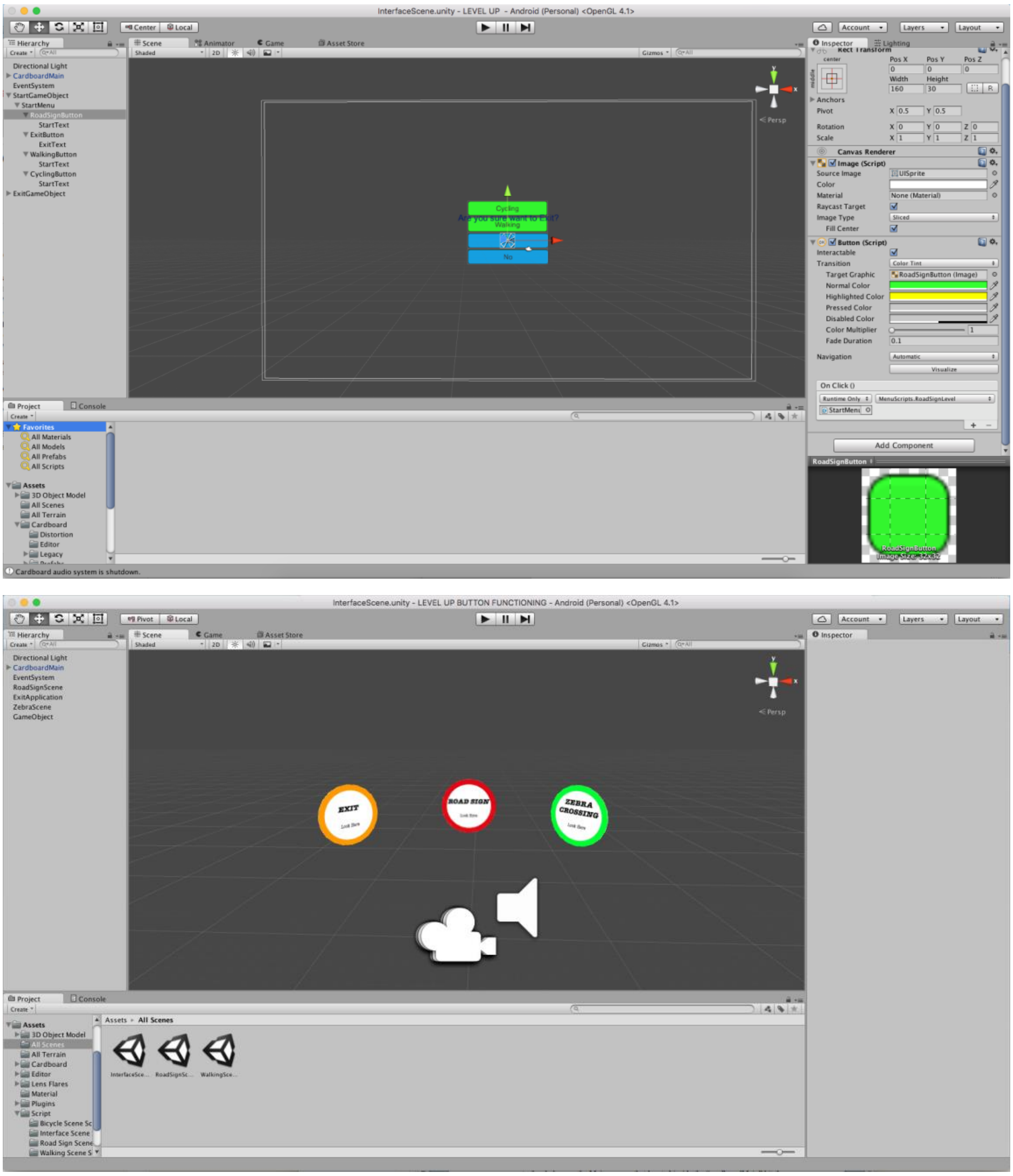
```
using UnityEngine;
using System.Collections;
using UnityEngine.SceneManagement;

public class RedialCutoutMenuZebra : MonoBehaviour {

    // How long to look at Menu Item before taking action
    public float timerDuration = 2f;
```

```
// This value will count down from the duration
private float lookTimer = 0f;

// My renderer so I can set _Cutoff value
private Renderer myRenderer;

// Box Collider
private BoxCollider myCollider;

// Is player looking at me?
private bool isLookedAt = false;

// MonoBehaviour Start
void Start() {
    // My Collider
    myCollider = GetComponent<BoxCollider>();
    // Get my Renderer
    myRenderer = GetComponent<Renderer>();
    // Set cutoff
    myRenderer.material.SetFloat("_Cutoff", 0f);
}

// MonoBehaviour Update
void Update() {
    // While player is looking at me
    if (isLookedAt) {
        // Reduce Timer
        lookTimer += Time.deltaTime;

        // Set cutoff value on material to value between 0 and 1
        myRenderer.material.SetFloat("_Cutoff", lookTimer / timerDuration);
```

```
            if (lookTimer > timerDuration) {
                // Reset timer
                lookTimer = 0f;

                // disable collider
                myCollider.enabled = false;

                // Do something
                Debug.Log("BUTTON HAS BEEN SELECTED!");
                SceneManager.LoadScene (2);
                // Disappear
                gameObject.SetActive(false);
            }
        }  else {
            // Reset Timer
            lookTimer = 0f;
            // Reset Cutoff
            myRenderer.material.SetFloat("_Cutoff", 0f);
        }
    }

    // Google Cardboard Gaze
    public void SetGazedAt(bool gazedAt) {
        // Set the local bool to the one passed from Event Trigger
        isLookedAt = gazedAt;
    }
}
```

Figure 4.6: C# Menu Script that used for the interface button.

Every button in this interface must have this script but every button has their owns function based on the every function (public void) in that script. To do that, back into Unity and select “RoadSignButton”. In the Inspector there has Button (script) partition.



InterfaceScene.unity - LEVEL UP - Android (Personal) <OpenGL 4.1>

Rect Transform: Pos X 0, Pos Y 0, Pos Z 0; Width 160, Height 30; Pivot X 0.5, Y 0.5; Rotation X 0, Y 0, Z 0; Scale X 1, Y 1, Z 1

Image (Script): Source Image UISprite; Material None (Material); Raycast Target ☑; Image Type Sliced; Fill Center ☑

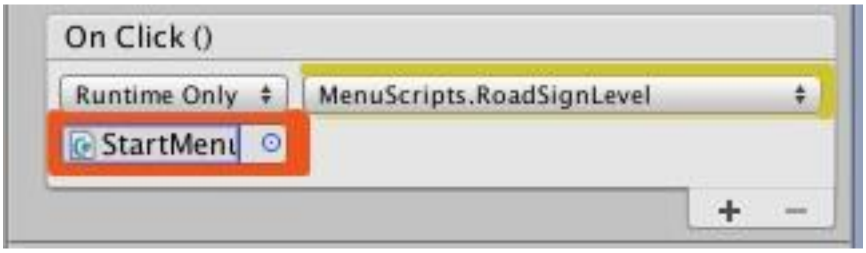
Button (Script): Interactable ☑; Transition Color Tint; Target Graphic RoadSignButton (Image); Normal Color 🟩; Highlighted Color 🟨; Color Multiplier 1; Fade Duration 0.1; Navigation Automatic

On Click (): Runtime Only; MenuScripts.RoadSignLevel; StartMenu

InterfaceScene.unity - LEVEL UP BUTTON FUNCTIONING - Android (Personal) <OpenGL 4.1>

Figure 4.7: Button (Script) at the inspector Window

In the Button (script), at On Click () choose the script that already created then select the Function that the button need to do after the user click on it. Change the setting to the entire button that has to make sure the button functioning after clicking. Lastly, create the skybox on the Main camera that located inside the “cardboardMain” hit the add component then select skybox then choose image that suitable for the interface. All the image can be download from the Internet for free.



On Click ()

Runtime Only

MenuScripts.RoadSignLevel

StartMenu

Figure 4.8: On Click at the inspector Window

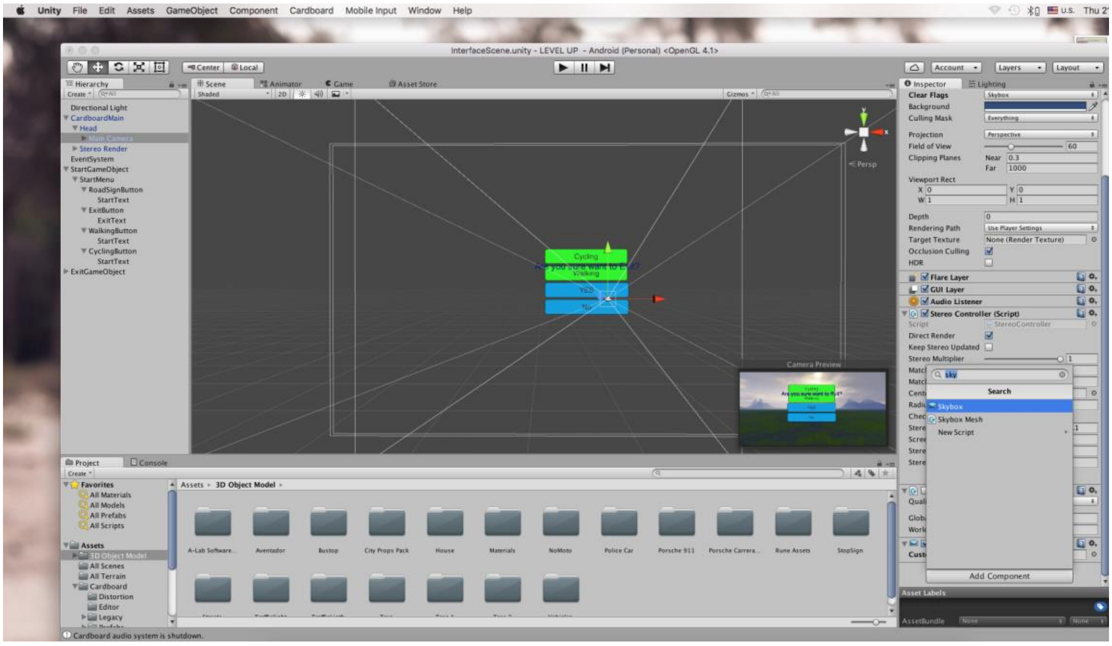

Unity File Edit Assets GameObject Component Cardboard Mobile Input Window Help
InterfaceScene.unity - LEVEL UP - Android (Personal) <OpenGL 4.1>
Hierarchy
Scene
Game
Asset Store
Inspector
Lighting
Search
Skybox
Skybox Mesh
New Script
Add Component
Project
Console
Assets › 3D Object Model

Figure 4.9: Setting the skybox view

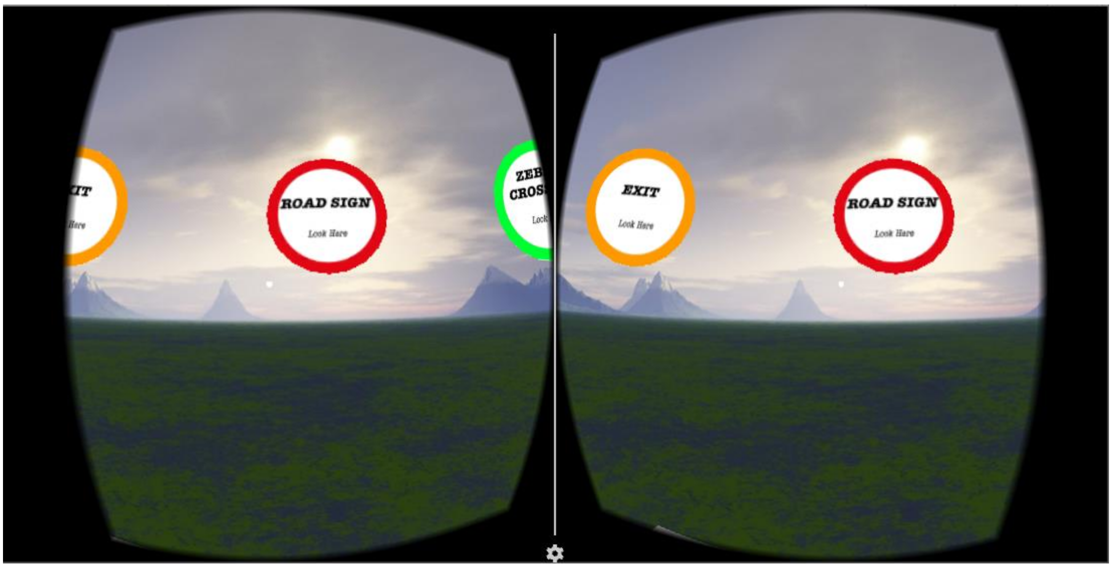


Figure 4.10: Output of the interface

This scene is about; give understanding to the user about road sign that familiar in the road. User needs to understand the meaning of the road sign. Thus, the road sign (3D Object) must be created. In this project, they have two ways to create the 3D object. The first way is create from Autodesk Maya and import that to the Unity and the second ways is from Unity it self. In this project it combine this two ways.

This project use Maya software for students that it can be download from Maya web site after registers it as students. After download and installed the software into computer the Maya software can be open by double click the Maya icon.

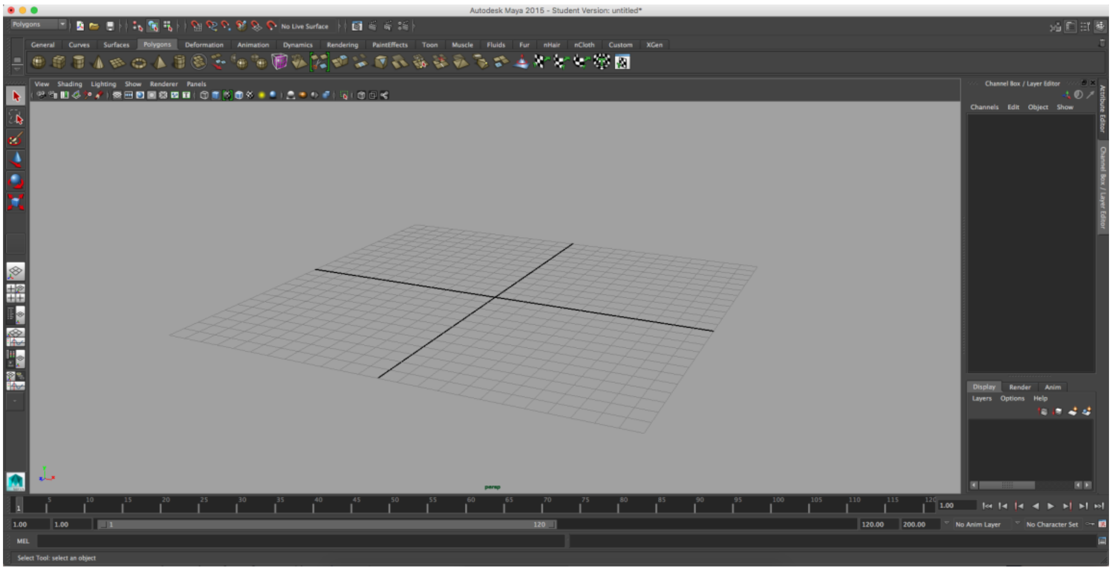
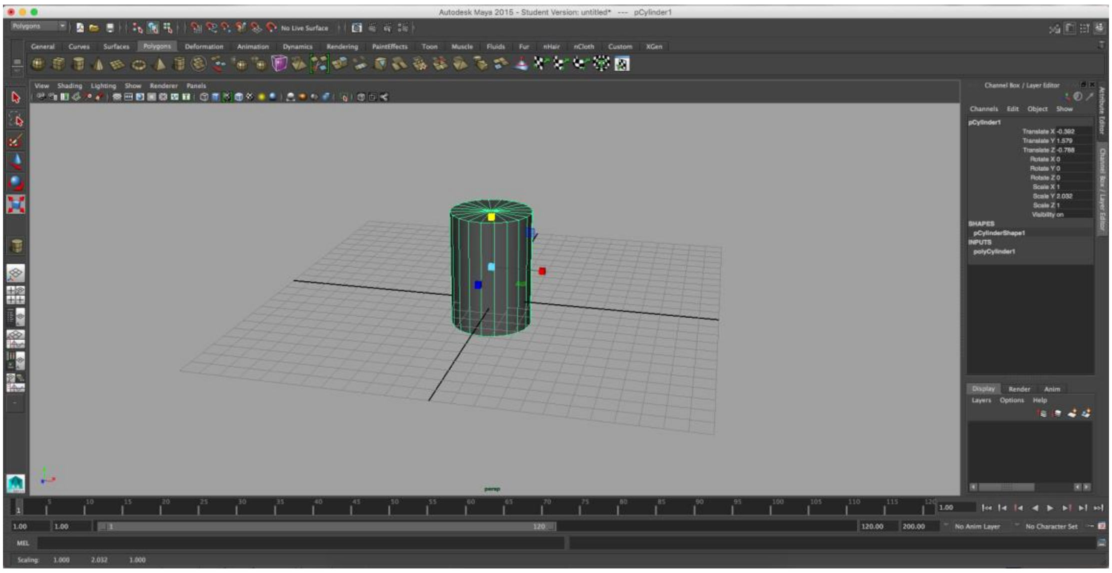


Figure 4.11: Maya Interface

To create the road signs from Maya for example, sign of motorcycle do not allow it has two parts which is the ions rods and the round iron for the picture of motorcycle with cross sign at the middle. To create the first part, use the polygons object in Maya menu then choose cylinder. Drag the mouse into the Maya scene at the same times press shift button in the keyboard to make sure the cylinder has the same size from x, y and z coordinate then click left button mouse and drag to draw the cylinder polygons. The images below shows example of cylinder polygons.



pCylinder1

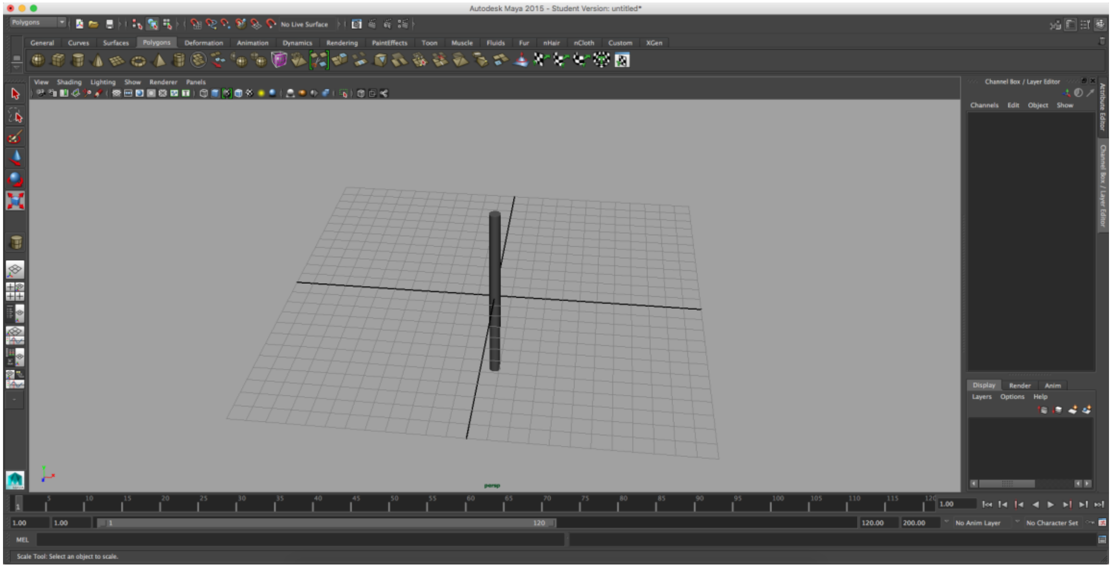
Translate X	-0.382
Translate Y	1.579
Translate Z	-0.788
Rotate X	0
Rotate Y	0
Rotate Z	0
Scale X	1
Scale Y	2.032
Scale Z	1
Visibility	on

SHAPES
pCylinderShape1
INPUTS
polyCylinder1

Scaling: 1.000 2.032 1.000

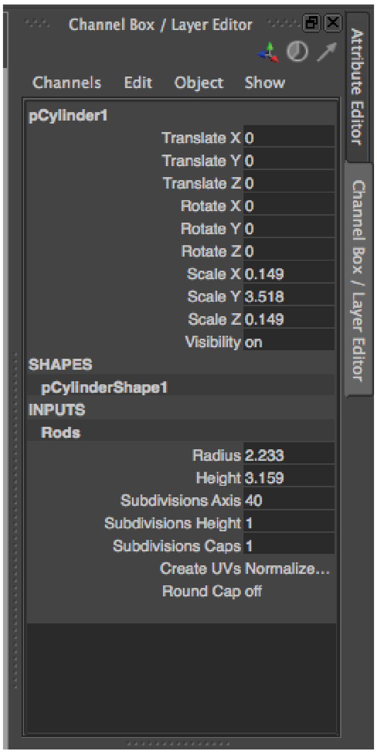
Figure 4.12: Creating cylinder polygons using Maya

In Maya the manipulation tool same goes to the Unity using W, E, and R letter on the keyboard to change the coordinate, rotation and scale / size of the object. For creating this sign it not take a lot of manipulation because the rods already in cylinder shape but only the size and the length of rods. Press R to change the size and length of cylinder until get the right length and size. Do the same things to create the polygons for the second part, which is the round ions, and resize it. To make sure the cylinder has a nice roundness increase the Subdivisions Axis at Channel box / layer Editor.



Scale Tool: Select an object to scale.

Figure 4.13: Cylinder polygons after resizing



pCylinder1

Translate X 0

Translate Y 0

Translate Z 0

Rotate X 0

Rotate Y 0

Rotate Z 0

Scale X 0.149

Scale Y 3.518

Scale Z 0.149

Visibility on

SHAPES

pCylinderShape1

INPUTS

Rods

Radius 2.233

Height 3.159

Subdivisions Axis 40

Subdivisions Height 1

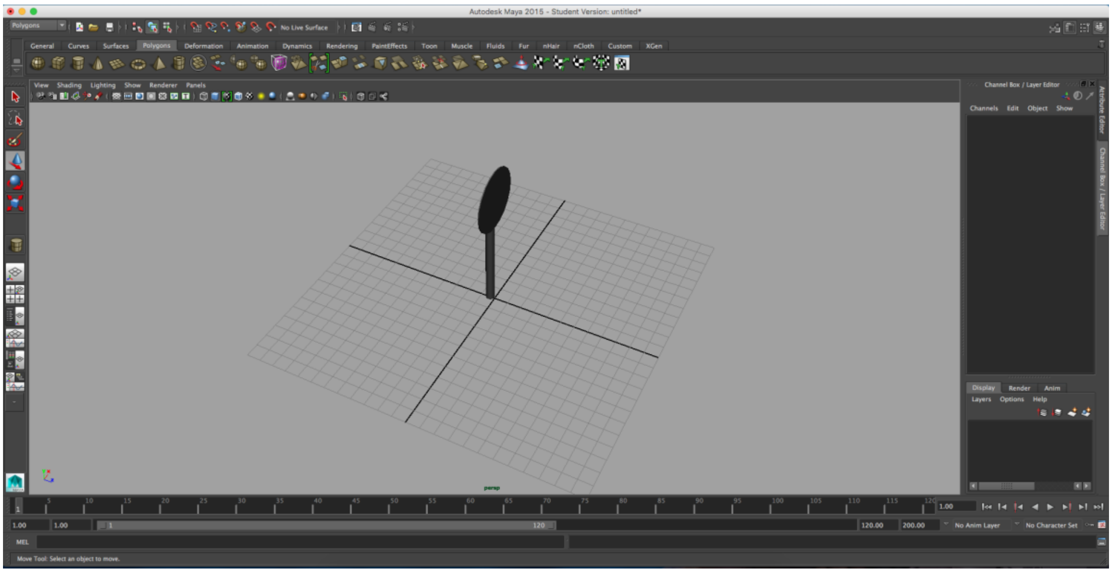
Subdivisions Caps 1

Create UVs Normalize…

Round Cap off

Figure 4.14: Channel box / layer Editor at Maya to change the Subdivisions Axis.

To make sure the two polygon combine each other do the grouping by select all the polygons then go to Edit > Group or press control + G at the keyboard or command + G for IMac computer. The outcome will be like the image below.



Move Tool: Select an object to move.

Figure 4.15: 3D object become after grouping with the first part and the second part

After completing it, now is the stage of make the texture mapping to the polygons only for the second part the rounds polygons. To do that, select the object that need to be textured then right click and hold it and choose face. After that right clicks and hold to the face that need to be texture choose assign favourite material then choose Phong E. The attribute editor will appear at the right side of the Maya. At the attribute editor go to colour the click on the small square coloured by black and white then choose file. At the attribute editor will appear “Image Name” then click at the folder icon then choose the image location that want to use it to be a texture.

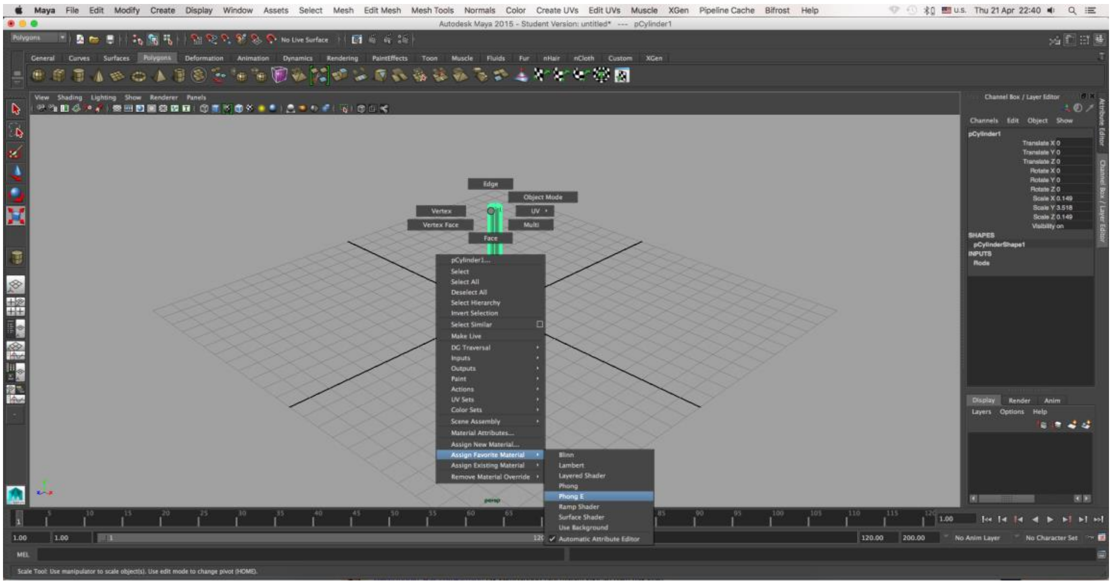


Figure 4.16: Assign the Phong E material to the 3D object

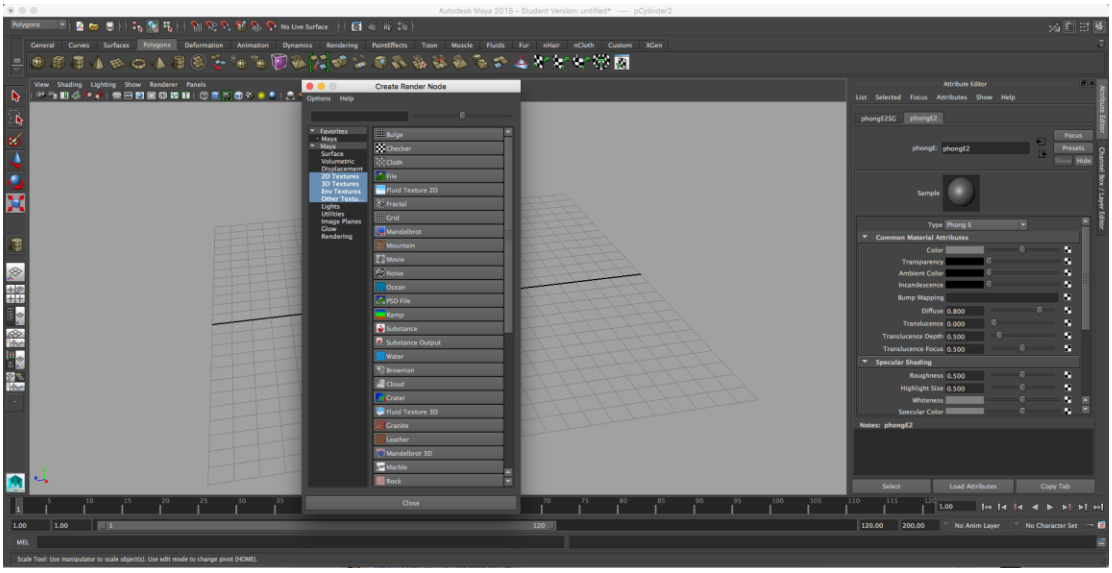


Figure 4.17: Apply texture on the Object using Phong E and Image

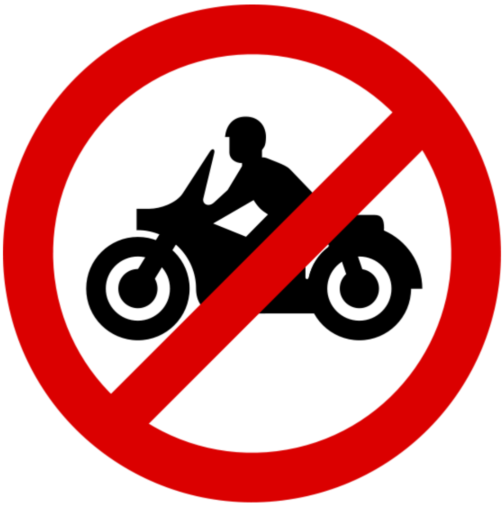
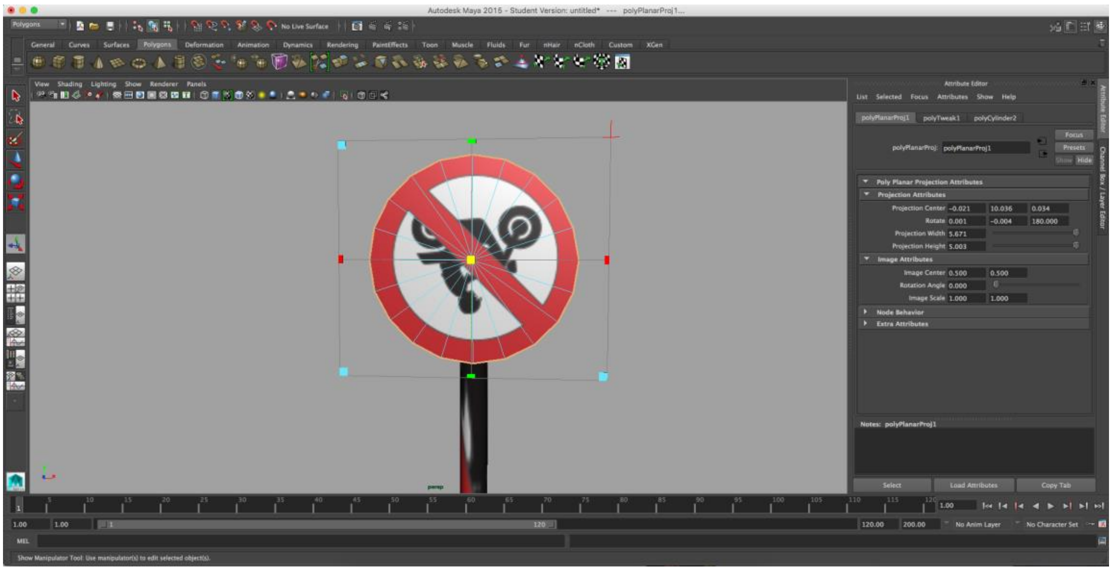


Figure 4.18: Image that used to make a texture mapping to the polygon



polyPlanarProj: polyPlanarProj1

Poly Planar Projection Attributes

Projection Attributes

Projection Center	-0.021	10.036	0.034
Rotate	0.001	-0.004	180.000
Projection Width	5.671		
Projection Height	5.003		

Image Attributes

Image Center	0.500	0.500
Rotation Angle	0.000	
Image Scale	1.000	1.000

Node Behavior

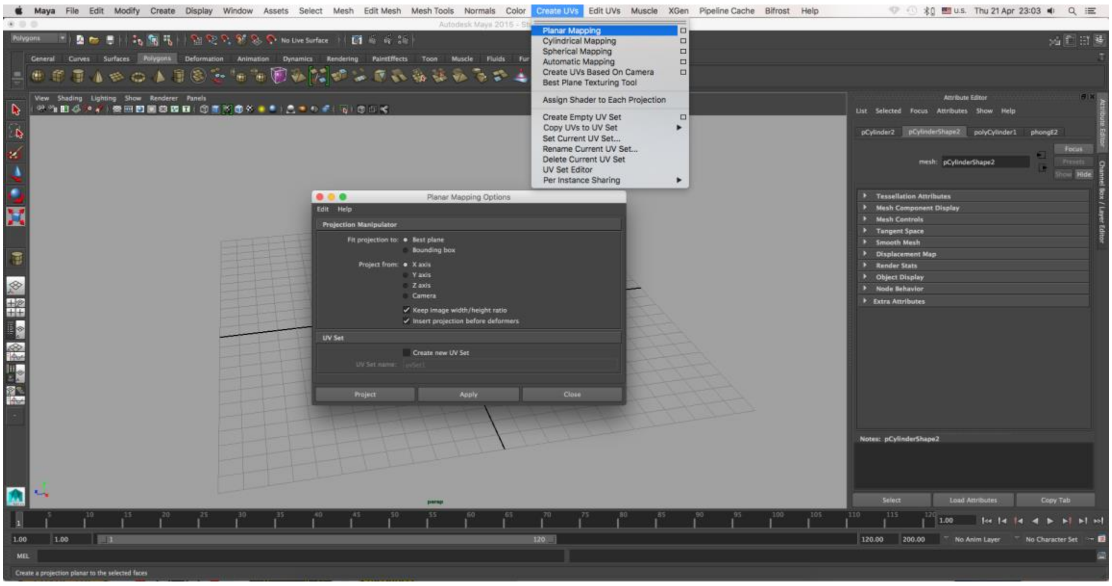
Extra Attributes

Notes: polyPlanarProj1

Show Manipulator Tool: Use manipulator(s) to edit selected object(s).

Figure 4.19: Polygon that has texture mapping on it

After done the texture mapping, sometimes the texture position and rotation does not match with the polygons. So, it is need to reposition by simply go to the Create UVs at the top menu of Maya the choose the planar mapping but click on the small square at the right side of planar mapping the click project. After finished the 3D object has complete been create. Then the 3D object can be save or export in Maya Binary format of FBX format. This format can be save into Unity Project that has created into the asset folder or save in into the computer and drag it into the Unity while development process anytime if the object needed. Below is the example image how to select the planar mapping and the image of complete 3D object has been done.



Planar Mapping Options

Projection Manipulator

Fit projection to: ● Best plane ○ Bounding box

Project from: ● X axis ○ Y axis ○ Z axis ○ Camera

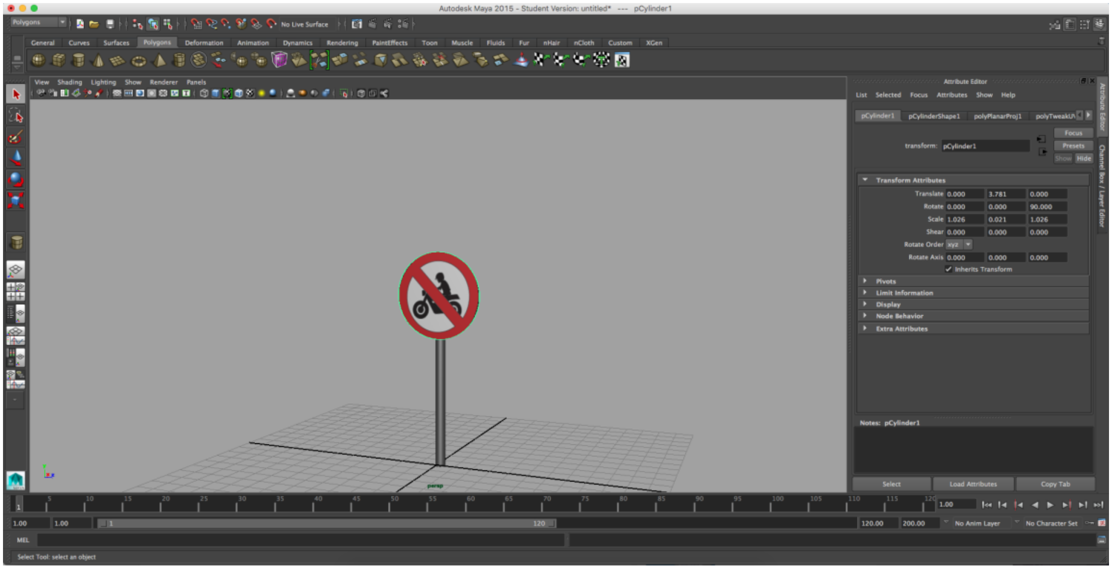
☑ Keep image width/height ratio

☑ Insert projection before deformers

UV Set

□ Create new UV Set

Figure 4.20: Used the planar mapping to reposition and rotation of the image



transform: pCylinder1

Transform Attributes

Translate	0.000	3.781	0.000
Rotate	0.000	0.000	90.000
Scale	1.026	0.021	1.026
Shear	0.000	0.000	0.000
Rotate Order	xyz		
Rotate Axis	0.000	0.000	0.000

☑ Inherits Transform

Figure 4.21: 3D object has been done creating

The second way to create the 3D object is from the unity it self. Using Unity the 3D object also can be created. To create the 3D object simply go to hierarchy window > right click > 3D object > cylinder. Then use the W, E, and R letter on the keyboard to change the coordinate, rotation and scale / size of the object. Then create another part of the sign. To make a group with the two cylinder jus simply drag the round cylinder into the rod cylinder.

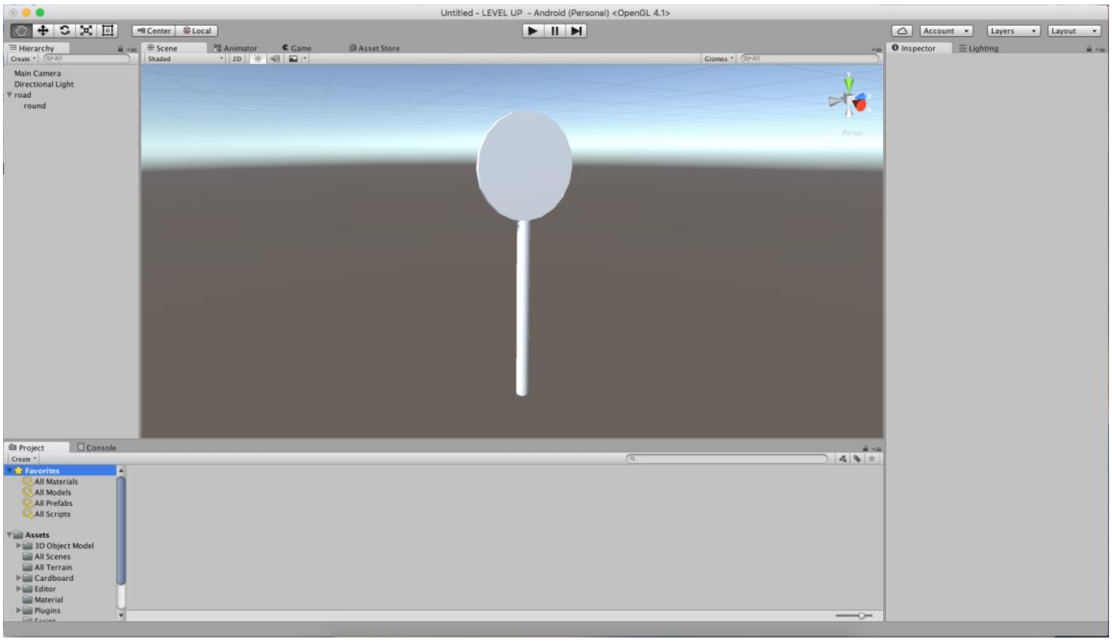


Figure 4.22: Create 3D object using Unity

To apply the texture mapping go to the project window right click > create > material. Then rename the material with the suitable name then click on it to assign the image. At the Inspector window choose the albedo then select the image. Drag the material into the object that needs to be texture. Create another material for the rod texture mapping, but the image creates from the Photoshop using black and white.

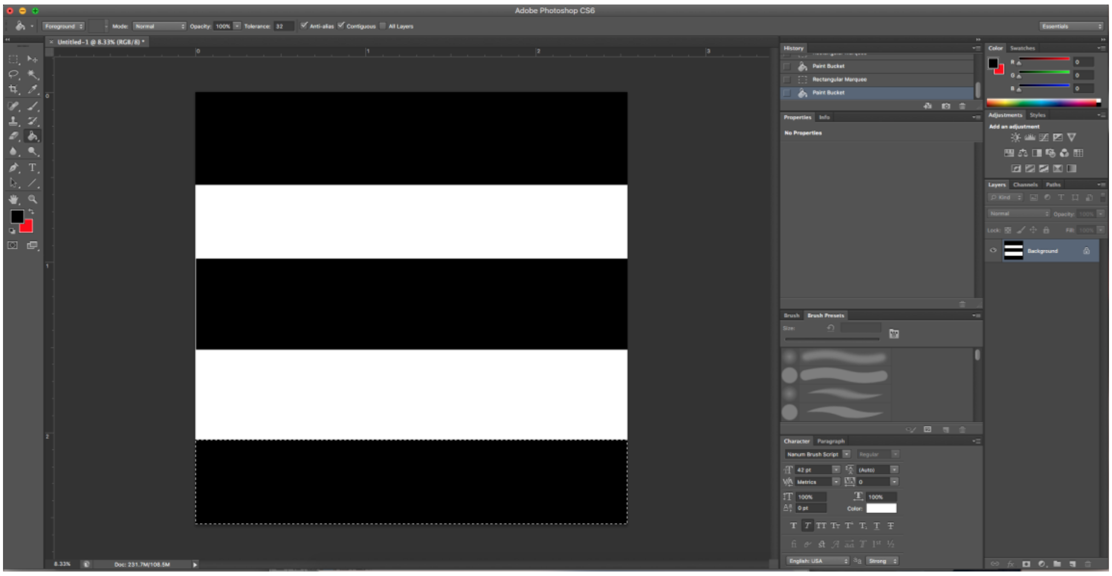


Figure 4.23: Create the texture mapping (image) for the object using Photoshop

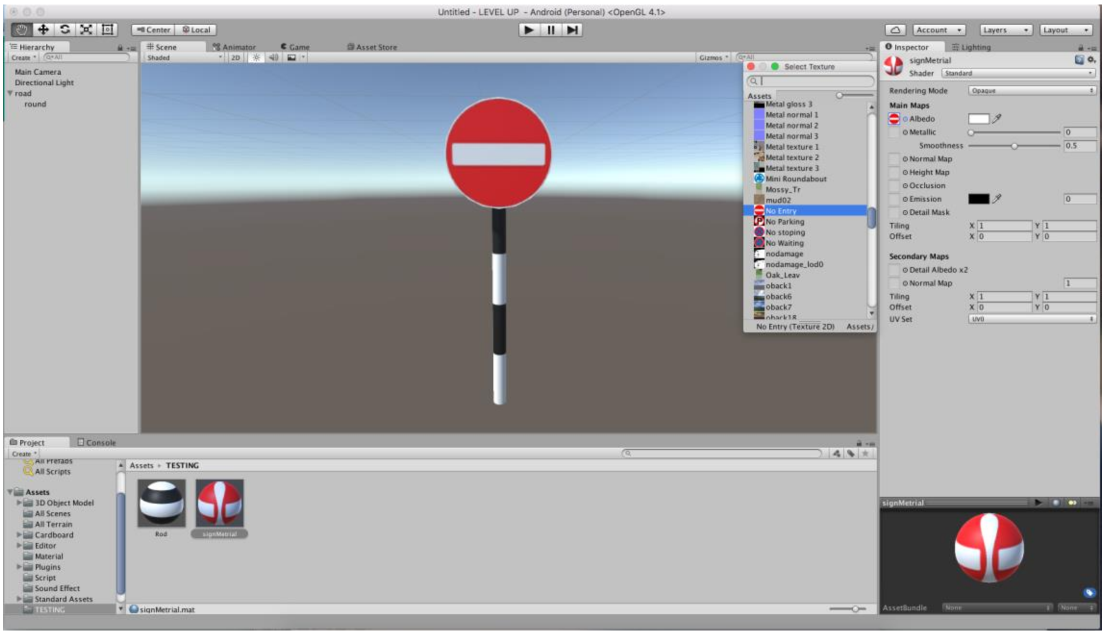


Figure 4.24: Complete road sign 3D object

After completing creates all the road signs that needs in this scene now start to create the scene inside Unity. In this scene, it is need to create a new scene but do not create a new project. This is because; it is need in one project but different scene to load the every scene to the button that already crated at the interface scene. At the drop down menu of Unity, click File and select new scene, Unity will create new scene in the same project. The unity interface will look like Figure 4.1. In this stage just follow the point 4.1: Importing SDK Development start from (**). After finished follow that stage, now time to create the base or ground of the scene. To create the ground, go to Unity drop down menu select Game Object > 3D Object > Terrain.

Select the Terrain at the hierarchy window then at the Inspector they has Terrain the place that user can make changing to the Terrain. To make the terrain look like ground, click to the Edit texture then select the image ground for example grass image. Then user can use the manipulation tool that has in Inspector for the terrain to make a mountain and anything that provide from the Unity. In this project, it just changes the texture and creates a mountain around the terrain. Then inside the unity they has the asset store that user can use to find the 3D object that needed in to the scene. This project use the asset store to find the 3D Object of road and all the road props like cone, roadblock stone, trashcan, tin can, and others.

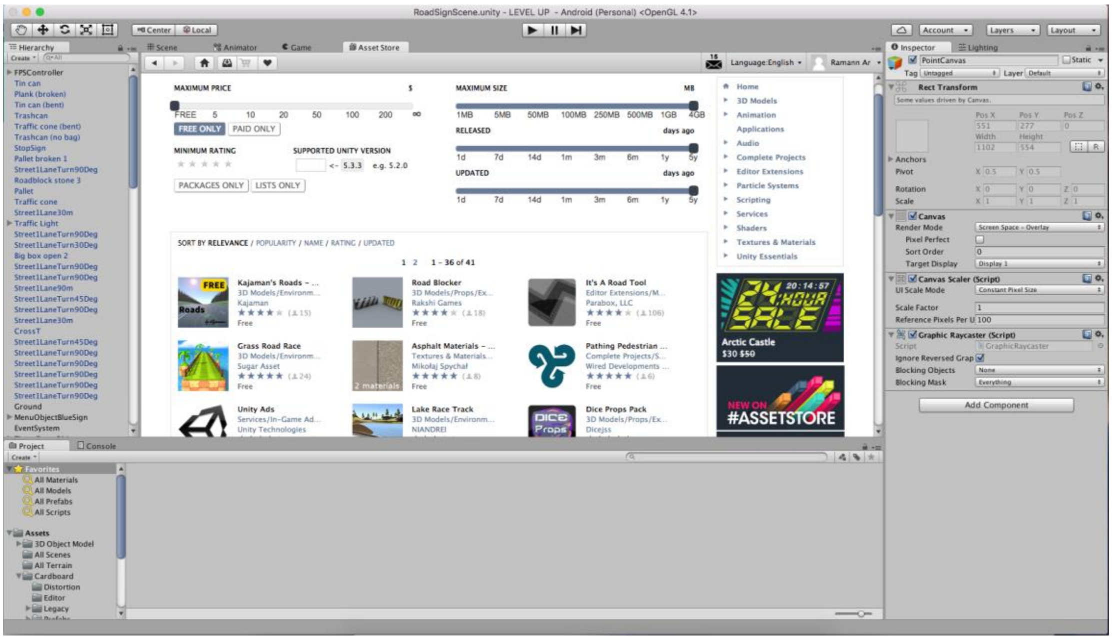


Figure 4.25: Asset store that Unity provides for user

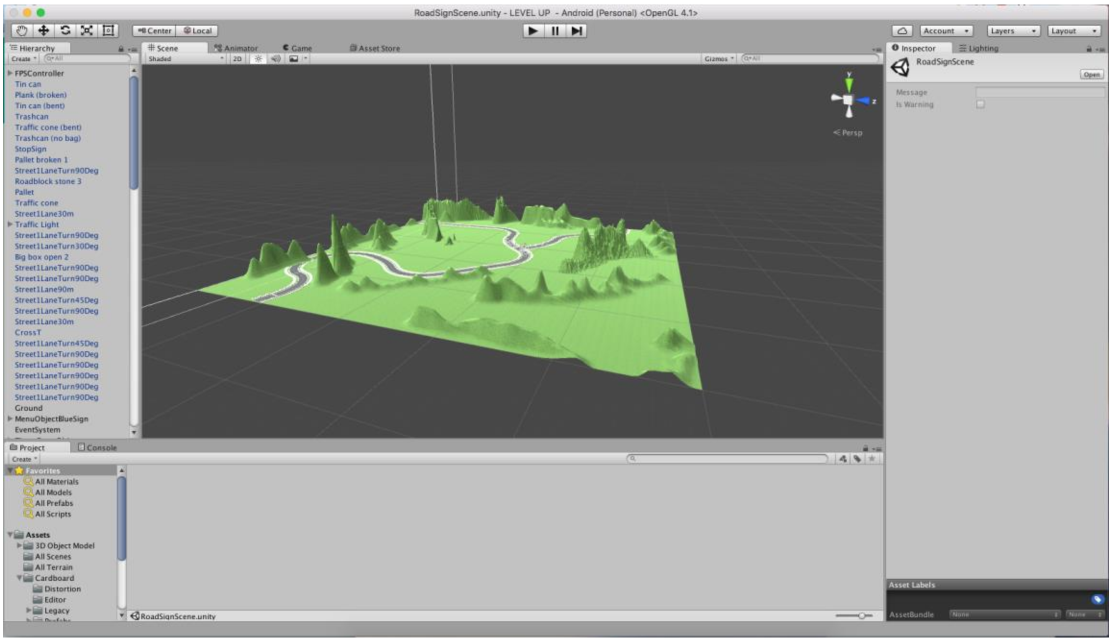


Figure 4.26: Image of complete developing the terrain.

All the 3D object located base on creativity of developer then all the 3D object that already created by Maya import into this scene. The entire 3D object has located at the asset folder in the Unity project. So, drag the 3D object into the scene the change the coordinate, rotation and scale / size of the 3D object.

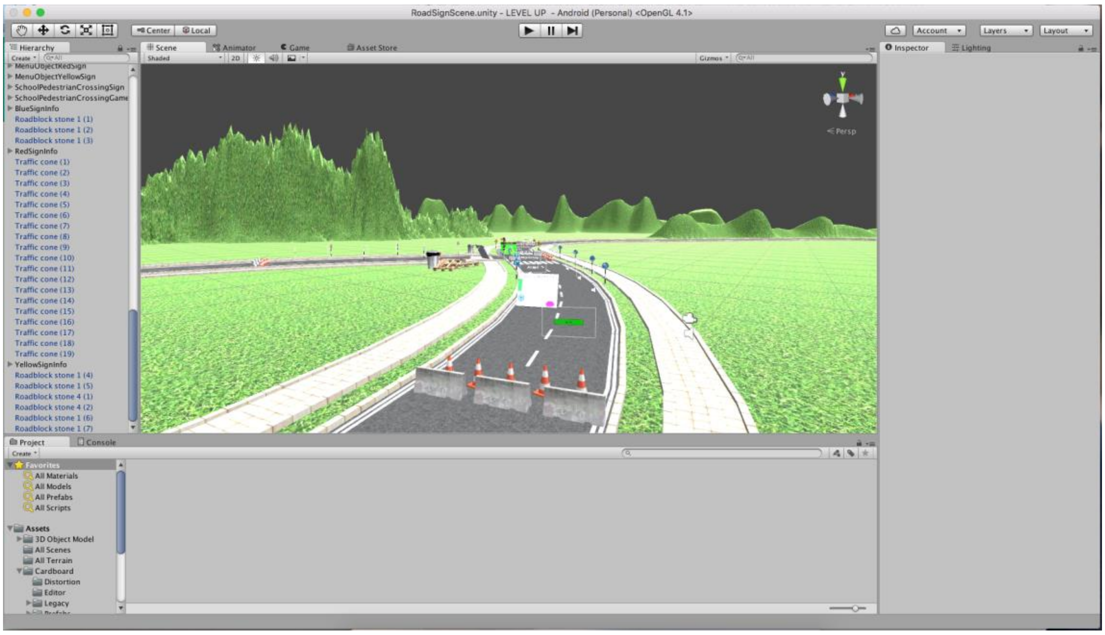
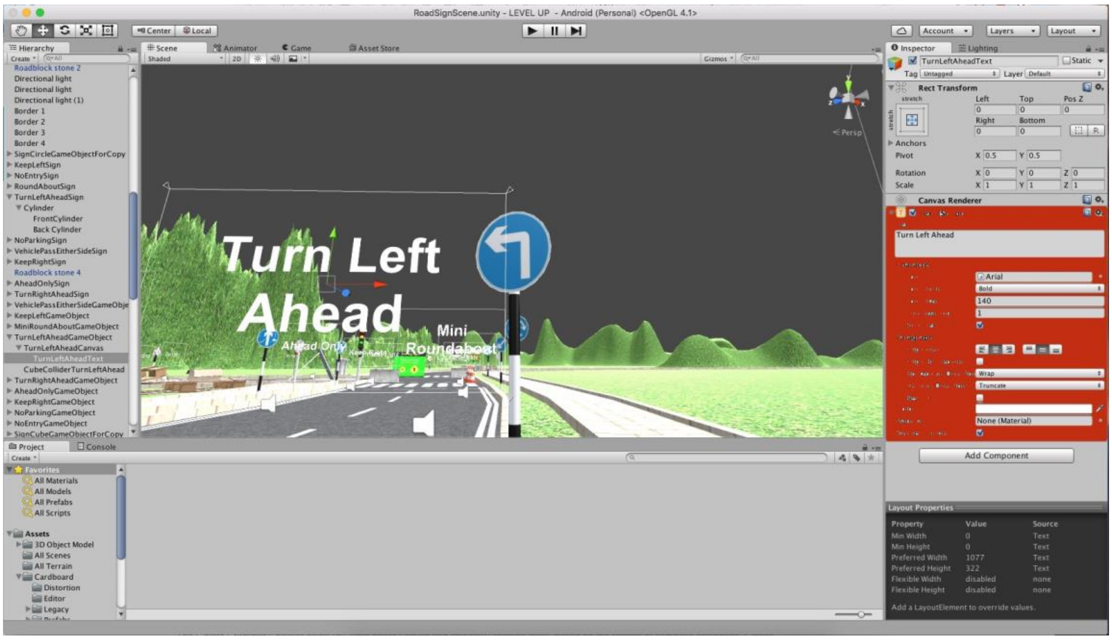


Figure 4.27: Completing import the 3D object into the scene and put in the right coordinate and position

After finished locating the 3D object (road signs) into the scene the road signs must be giving name based on the signs images. They have 17 road sings that must be giving name. To create the name of every road signs, canvas will be used. For example, create the name for turn left ahead road sign. Go to hierarchy window right click > create empty. Then rename it to "TurnLeftAheadGameObject". The name given must be related to the sign that need to give named. To create the canvas right click "TurnLeftAheadGameObject" > UI > Canvas then rename it to "TurnLeftAheadCanvas". After that right click at "TurnLeftAheadCanvas" > UI > text then rename it to "TurnLeftAheadText". Click the text then change the text at the text attribute at the inspector. At the inspector the text can be manipulated like change the name, the font, colour and size. Repeat that process to the rest of the road signs.

However, to make this project more interactive to the users the name of the road signs is hidden. Only and only if the user go to near the road sign so the name will be appear at the side of the road signs. Thus, to do so applies one cube collider in front of the road signs. To create the box collider, right click the hierarchy windows > 3D object > Cube. Resize and locate the suitable position in front of the road signs. Then unclick the Mesh Renderer at the Inspector windows. Do the same things to the entire road signs.


RoadSignScene.unity - LEVEL UP - Android (Personal) <OpenGL 4.1>
Turn Left Ahead
Mini Roundabout
Ahead Only
Add Component

Inspector: ☑ TurnLeftAheadText □ Static
Tag Untagged Layer Default

Rect Transform
Left 0 Top 0 Pos Z 0
Right 0 Bottom 0
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Canvas Renderer

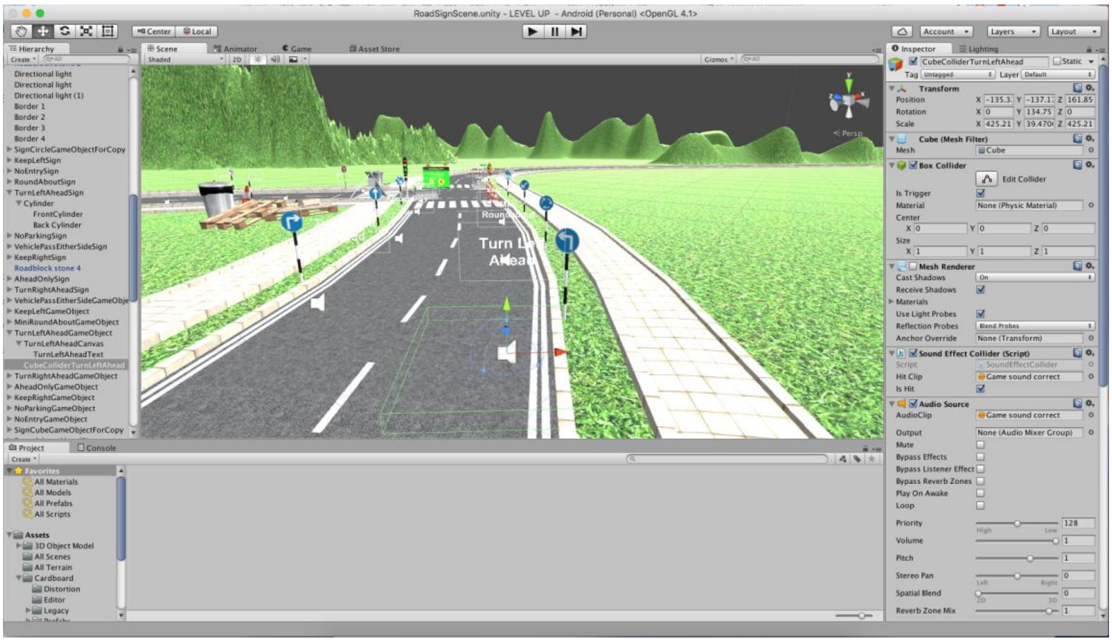
Turn Left Ahead
Arial
Bold
140
1
☑
Wrap
Truncate
None (Material)
☑

Layout Properties

Property	Value	Source
Min Width	0	Text
Min Height	0	Text
Preferred Width	1077	Text
Preferred Height	322	Text
Flexible Width	disabled	none
Flexible Height	disabled	none

Add a LayoutElement to override values.

Figure 4.28: Change the name of the text at the inspector


RoadSignScene.unity - LEVEL UP - Android (Personal) <OpenGL 4.1>
Turn Left Ahead
Roundabout

Inspector: ☑ CubeColliderTurnLeftAhead □ Static
Tag Untagged Layer Default

Transform
Position X -135.3 Y -137.1 Z 161.85
Rotation X 0 Y 134.75 Z 0
Scale X 425.21 Y 39.470 Z 425.21

Cube (Mesh Filter)
Mesh Cube

☑ Box Collider
Edit Collider
Is Trigger ☑
Material None (Physic Material)
Center X 0 Y 0 Z 0
Size X 1 Y 1 Z 1

□ Mesh Renderer
Cast Shadows On
Receive Shadows ☑
Materials
Use Light Probes ☑
Reflection Probes Blend Probes
Anchor Override None (Transform)

☑ Sound Effect Collider (Script)
Script SoundEffectCollider
Hit Clip Game sound correct
Is Hit ☑

☑ Audio Source
AudioClip Game sound correct
Output None (Audio Mixer Group)
Mute □
Bypass Effects □
Bypass Listener Effect □
Bypass Reverb Zones □
Play On Awake □
Loop □
Priority 128
Volume 1
Pitch 1
Stereo Pan 0
Spatial Blend 0
Reverb Zone Mix 1

Figure 4.29: Applying the box collider in front of the road signs then Unclick the mesh renderer

To make the box collider functioning the script must be applying into it. The idea is if the First person controller (FPS) or player hit the box collider the road sign name will appear then if the FPS do not hit the box collider the road sign name will hidden. At the same time the sound also applied if the FPS hit the box collider but only for one times then the road signs complete hit the collider will shows at the top right screen. The road signs complete will counter up if the FPS hit the box collider. But, the counter up only one times even the FPS hits the same box collider many times.

Before creating the script the FPS must be import into the scene. To do so go to the drop down menu of Unity assets > import package > Character. After importing the FPS go to folder in project window standard assets > character > first person controller > prefabs > FPS controller then drag it into the scene. Inside the FPScontroller at the hierarchy window delete the Main camera then drag in the cardboard main into the FPS.

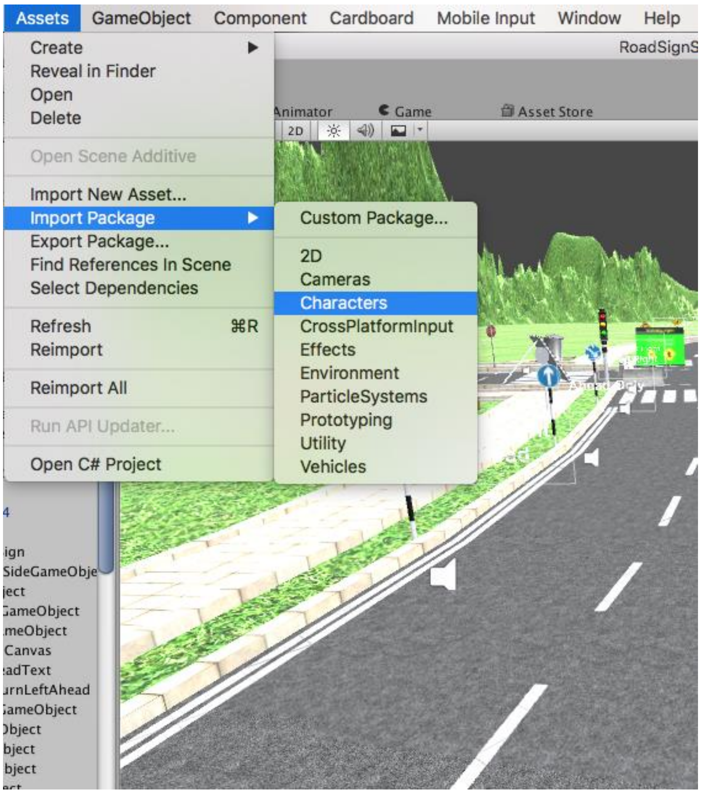


Figure 4.30: Importing the FPS into the scene.

After the entire 3D object, road signs and the FPS import into scene, now is creating the script for the box collider and the menu button. To create the menu button it is same way to create at the button at the interface scene. But the script that applies is different.

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;

public class FadeInOut : MonoBehaviour {

    public Text SignText;
    public float fadeSpeed = 5f;
    public bool entrance;
    public GameObject TextSignCanvas;

    public PointController pointController;
    public bool firstEnter = true;

    void Awake()
    {
        //undeadText = undeadCemetery.GetComponentInChildren<Text> ();
        SignText = TextSignCanvas.GetComponentInChildren<Text> ();
        SignText.color = Color.clear;
    }
    // Use this for initialization
    void Start ()
    {

    }

    // Update is called once per frame
    void Update ()
    {
        ColourChange();
    }
    void OnTriggerEnter (Collider other) {
        entrance = true;
        marks ();
    }

    void OnTriggerExit (Collider other)
 {
        entrance = false;
    }

    void ColourChange ()
    {
        if (entrance)
        {
            SignText.color = Color.Lerp (SignText.color, Color.white, fadeSpeed * Time
.deltaTime);
        }
        else if (!entrance)
        {
            SignText.color = Color.Lerp (SignText.color, Color.clear, fadeSpeed * Time.
deltaTime);
        }
    }

    void marks()
    {
        if (firstEnter)
        {
            pointController.Score += 1;
            firstEnter = false;
        }
        else if (firstEnter)
        {
            pointController.Score += 0;
        }
    }
}
```

Figure 4.31: FadeInOut Script that apply into the box collider.

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;

public class PointController : MonoBehaviour {
    int score = 0;
    public Text pointText;
    public Text completeText;
    public float fadeSpeed = 5f;
    public int countdown;

    // Use this for initialization
    public int Score
    {
        get { return score;}
        set
        {
            score = value;
            UpdateUI () ;
        }
    }

    void Start ()
    {
        completeText.text = "";
        UpdateUI ();
    }
    // Update is called once per frame
    void UpdateUI ()
    {
        pointText.text = string.Format ("Signs:{0}/17", score);
        if (Score >= 17)
        {
            countdown = 5;
            endText ();
        }
    }

    void endText() {
        if (countdown > 0) {
            countdown--;
            completeText.text = "Complete";
            Invoke("endText",1);
        }
        else
        {
            completeText.text = "";
        }
    }
}
```

Figure 4.32: PointController Script that becoming the counter object the hold the value of point / road signs that user view.

```
#pragma strict
var hitClip:AudioClip;
var isHit = true;
```

```
function OnTriggerEnter(o:Collider)
{
   //isHit = true;
   if(isHit == true)
   {
      playHit();
   }
}

function OnTriggerExit(o:Collider)
{
   isHit = false;
}

function playHit()
{
      GetComponent.<AudioSource>().Play();
}

function Start ()
{


}

function Update ()
{


}
```

Figure 4.33: SoundEffect Script that apply into the box collider to make sound if the FPS hit the box collider.

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;

public class Timer : MonoBehaviour {
    public Text timerText;
    private float startTime;
    // Use this for initialization
    void Start ()
   {
        startTime = Time.time;
    }
    // Update is called once per frame
    void Update () {
        float t = Time.time - startTime;
        string minutes = ((int)t / 60).ToString ();
        string second = (t % 60).ToString ("f2");
        timerText.text = minutes + ":" + second;
    }
}
```

Figure 4.34: Timer Script that apply into the FPS to user view the time taken.

```
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
using System.Collections;

public class BackToMenu : MonoBehaviour {
    public Button MenuButton;
    // Use this for initialization
    void Start () {
        MenuButton = MenuButton.GetComponent<Button> ();
    }
    public void BackMenu()
    {
        Application.LoadLevel (0);


    }
    // Update is called once per frame
    void Update () {
    }
}
```

Figure 4.35: MenuButton Script that apply into the button to user going back to the interface scene.

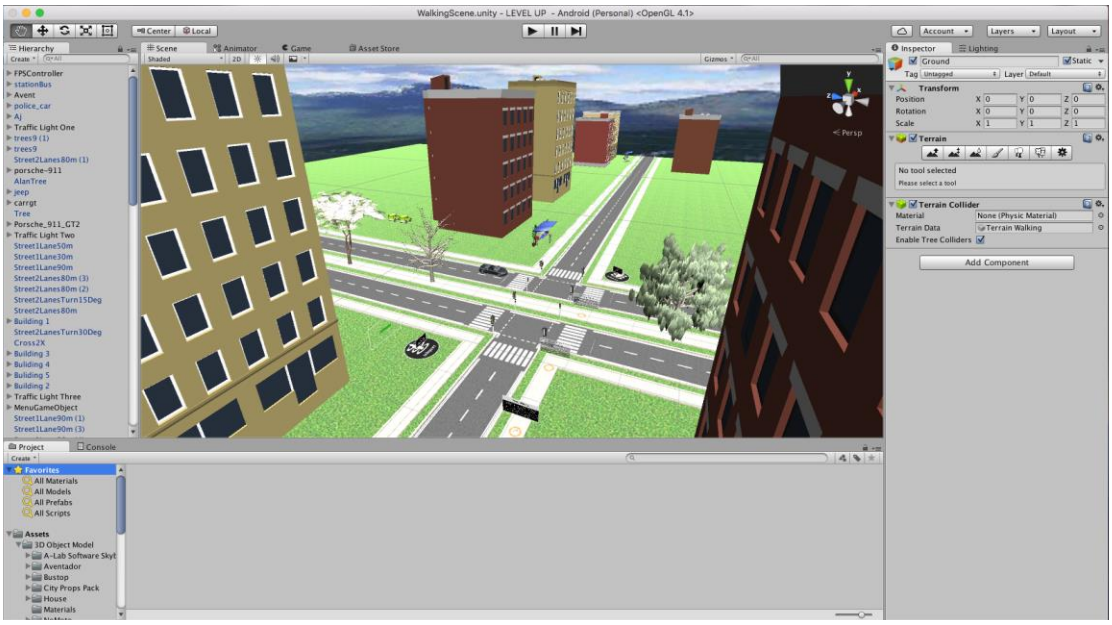
In this scene, it is about how the user can cross the road and this scene will give information and guide how the user cross the roads by using zebra crossing and non zebra crossing. This scene also has traffic light that user can follow to crossing the road. To get the information the user must stand at the top of information signs. Then there also has teleport that user can go to the place that non zebra crossing without walking. This is because; the location in that scene is far to each other.

The technique to develop this scene in this stages almost the same with the roads signs scene. Before creating the scene, the 3D object must be created. So, create the 3D object using Maya or Unity. For example that 3D object need to be created is Roads, Cars, Buildings, Traffic light, and others. However, the 3D object also can be finds at Assets Store Unity or from the tf3dm.com to get free 3D object.

After creating the 3D object, start to create the scene inside Unity. In this scene, it is need to create a new scene but do not create a new project. This is because; it is need in one project but different scene to load the every scene to the button that already crated at the interface scene. This steps same to the roads signs scene before. At the drop down menu of Unity, click File and select new scene, Unity will create new scene in the same project. The unity interface will look like Figure 4.1. In this stage just follow the point 4.1: Importing SDK Development start from (**). After finished follow that stage, now time to create the base or ground of the scene. To create the ground, go to Unity drop down menu select Game Object > 3D Object > Terrain.

The command technique to create the Terrain is same with before, same way to create terrain in the road signs scene, it own creativity how to develop the nice scene and follow the storyboard of the project. Select the Terrain at the hierarchy window then at the Inspector they has Terrain the place that user can make changing to the Terrain. To make the terrain look like ground, click to the Edit texture then select the image ground for example grass image. Then user can use the manipulation tool that has in Inspector for the terrain to make a mountain and anything that provide from the Unity. In this project, it just changes the texture of the terrain.

All the 3D object located base on creativity of developer then all the 3D object that already created by Maya import into this scene. The entire 3D object has located at the asset folder in the Unity project. So, drag the 3D object into the scene the change the coordinate, rotation and scale / size of the 3D object. Figure below shows how the 3D objects arranged in this scene.



☑ Ground ☑ Static

Tag Untagged Layer Default

Transform

Position X 0 Y 0 Z 0

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

☑ Terrain

No tool selected

Please select a tool

☑ Terrain Collider

Material None (Physic Material)

Terrain Data Terrain Walking

Enable Tree Colliders ☑

Add Component

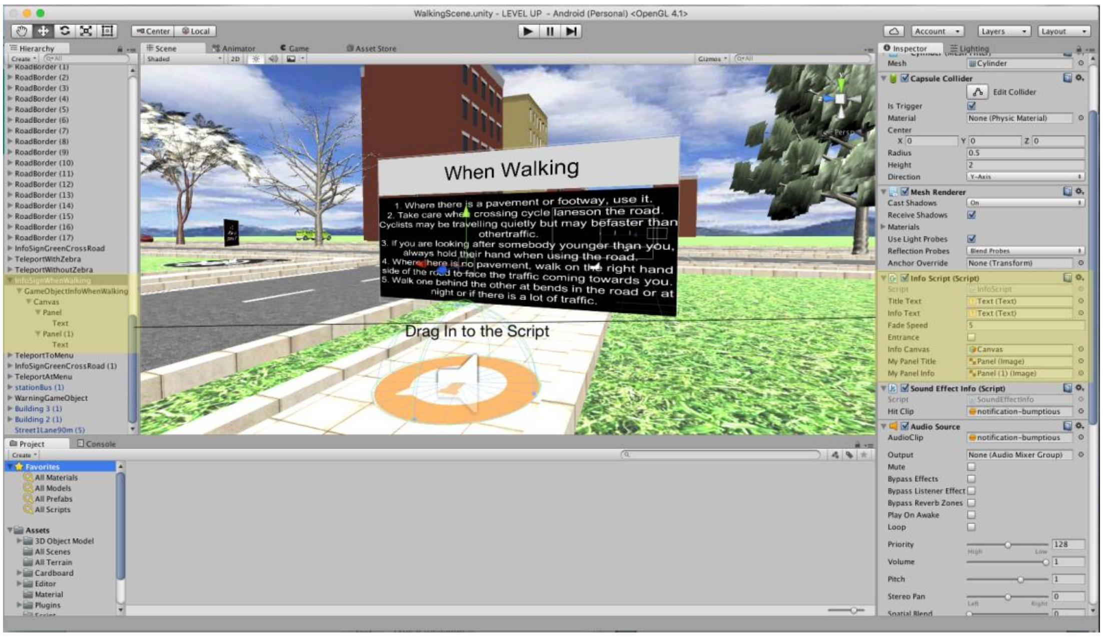
Figure 4.36: Arrangement of 3D objects in this scene

After getting the position of 3D object, now time for make some script for the 3D object. The user/player must hit the info images to get the info from fade in/out canvas. The idea is; the canvas will make distraction if the canvas appears along the time the scene running. So, apply the script below to the info 3D Object/image/box collider. Drag this script to the box collider then drag the canvas into the box collider slot that appears at inspector window.

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;

public class InfoScript : MonoBehaviour {

    public Text TitleText;
    public Text InfoText;
    public float fadeSpeed = 5f;
    public bool entrance;
    public GameObject InfoCanvas;

    public Image myPanelTitle;
    public Image myPanelInfo;

    // Use this for initialization
    void Start ()
    {
        //undeadText = undeadCemetery.GetComponentInChildren<Text> ();
        TitleText = InfoCanvas.GetComponentInChildren<Text> ();
        TitleText.color = Color.clear;
    }

    // Update is called once per frame
    void Update ()
    {
        ColourChange();
    }

    void OnTriggerEnter (Collider other) {
        entrance = true;
    }

    void OnTriggerExit (Collider other) {
        entrance = false;
    }



    void ColourChange ()
    {

        if (entrance)
        {
            TitleText.color = Color.Lerp (TitleText.color, Color.black, fadeSpeed * Time
.deltaTime);
            InfoText.color = Color.Lerp (InfoText.color, Color.white, fadeSpeed * Time.
deltaTime);
            myPanelTitle.color = Color.Lerp (myPanelTitle.color, Color.white, fadeSpee
d * Time.deltaTime);
            myPanelInfo.color = Color.Lerp (myPanelInfo.color, Color.black, fadeSpeed
* Time.deltaTime);
        }

        else if (!entrance)
        {
            TitleText.color = Color.Lerp (TitleText.color, Color.clear, fadeSpeed * Time.
deltaTime);
```

```
            InfoText.color = Color.Lerp (InfoText.color, Color.clear, fadeSpeed * Time.deltaTime);
            myPanelTitle.color = Color.Lerp (myPanelTitle.color, Color.clear, fadeSpeed * Time.deltaTime);
            myPanelInfo.color = Color.Lerp (myPanelInfo.color, Color.clear, fadeSpeed * Time.deltaTime);
        }
    }
}
```

Figure 4.37: Script that applies to the box collider to make the canvas fade in/out.



When Walking

1. Where there is a pavement or footway, use it.
2. Take care when crossing cycle laneson the road. Cyclists may be travelling quietly but may befaster than othertraffic.
3. If you are looking after somebody younger than you, always hold their hand when using the road.
4. Where there is no pavement, walk on the right hand side of the road to face the traffic coming towards you.
5. Walk one behind the other at bends in the road or at night or if there is a lot of traffic.

Mesh: Cylinder

☑ Capsule Collider
Edit Collider
Is Trigger: ☑
Material: None (Physic Material)
Center: X 0 Y 0 Z 0
Radius: 0.5
Height: 2
Direction: Y-Axis

☑ Mesh Renderer
Cast Shadows: On
Receive Shadows: ☑
Materials
Use Light Probes: ☑
Reflection Probes: Blend Probes
Anchor Override: None (Transform)

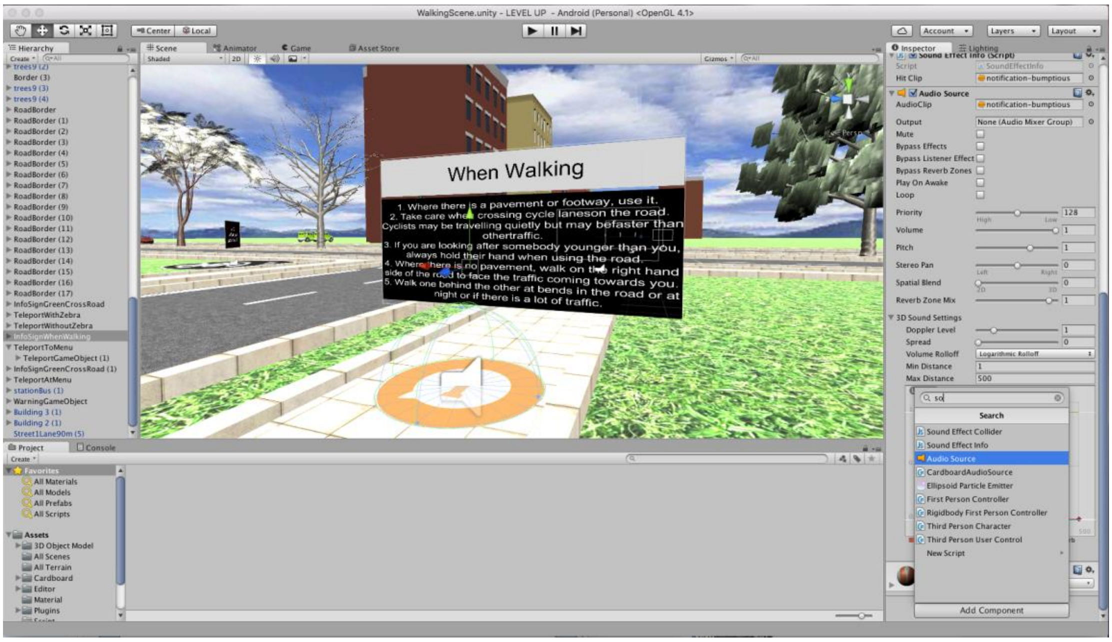
☑ Info Script (Script)
Script: InfoScript
Title Text: Text (Text)
Info Text: Text (Text)
Fade Speed: 5
Entrance: □
Info Canvas: Canvas
My Panel Title: Panel (Image)
My Panel Info: Panel (1) (Image)

☑ Sound Effect Info (Script)
Script: SoundEffectInfo
Hit Clip: notification-bumptious

☑ Audio Source
AudioClip: notification-bumptious
Output: None (Audio Mixer Group)
Mute: □
Bypass Effects: □
Bypass Listener Effect: □
Bypass Reverb Zones: □
Play On Awake: □
Loop: □
Priority: 128
Volume: 1
Pitch: 1
Stereo Pan: 0
Spatial Blend: 0

Figure 4.38: Text and Panel must be drag in into the slot that created in the script at inspector windows.

The sound effect also applies to the box collider to make some sound effect if the user/player hit the box collider. To do that, the script also needed. The script that applies inside the box collider is shown below. Before apply the script the sound source must be added into the box collider. Select the box collider; at the inspector click button add component then select sound source. Then at the first properties, which are, audio clip click the small play button at the right corner then select sound clip that want to use it. The sound clip must be in the .mp3 or .wav format.



When Walking

1. Where there is a pavement or footway, use it.
2. Take care when crossing cycle laneson the road. Cyclists may be travelling quietly but may befaster than othertraffic.
3. If you are looking after somebody younger than you, always hold their hand when using the road.
4. Where there is no pavement, walk on the right hand side of the road to face the traffic coming towards you.
5. Walk one behind the other at bends in the road or at night or if there is a lot of traffic.

Sound Effect Info (Script)
Script: SoundEffectInfo
Hit Clip: notification-bumptious

☑ Audio Source
AudioClip: notification-bumptious
Output: None (Audio Mixer Group)
Mute: □
Bypass Effects: □
Bypass Listener Effect: □
Bypass Reverb Zones: □
Play On Awake: □
Loop: □
Priority: 128
Volume: 1
Pitch: 1
Stereo Pan: 0
Spatial Blend: 0
Reverb Zone Mix: 1
3D Sound Settings
Doppler Level: 1
Spread: 0
Volume Rolloff: Logarithmic Rolloff
Min Distance: 1
Max Distance: 500

Search: so
Sound Effect Collider
Sound Effect Info
Audio Source
CardboardAudioSource
Ellipsoid Particle Emitter
First Person Controller
Rigidbody First Person Controller
Third Person Character
Third Person User Control
New Script

Figure 4.39: Add sound source into the box collider then select the sound clip that want to use it from the computer.

```
#pragma strict
var hitClip:AudioClip;


function OnTriggerEnter(o:Collider)
{
   GetComponent.<AudioSource>().Play();
}
function Start ()
{


}
function Update ()
 {
}
```

Figure 4.40: Script that apply into the box collider.

To make the teleport create any 3D object that need to use, for this scene is cylinder. Go to the hierarchy window right click > 3D object > cylinder. Resize and re name the cylinder to “Teleport 1”. Apply texture mapping into the cylinder then copy it to make two copies and rename it to “Teleport 2”. Then create and apply script to the both Teleport 3D Object to make sure when the user/player hit the Teleport 1 it will jump to teleport 2 then same goes if the user/player hit the teleport 2 it will go to the teleport 1.

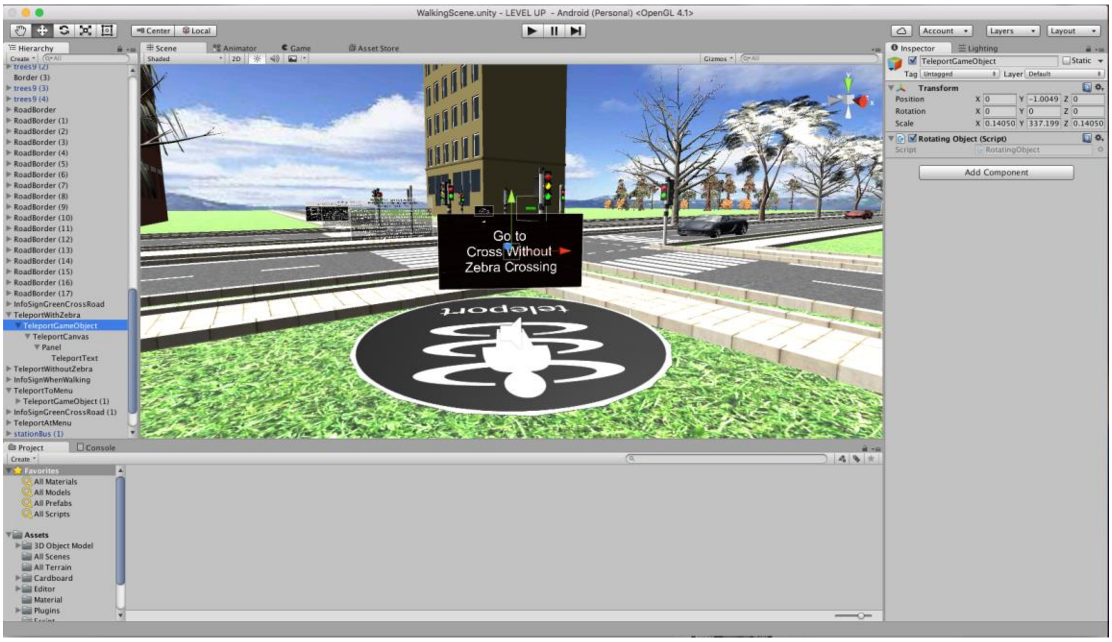
```
using UnityEngine;
using System.Collections;

public class TeleportPad : MonoBehaviour {

   public int code;
   float disableTimer = 0;

   void Update ()
   {
      if (disableTimer > 0)
      {
         disableTimer -= Time.deltaTime;
      }
   }

   void OnTriggerEnter(Collider collider)
   {
      if (collider.gameObject.name == "FPSController" && disableTimer <= 0)
      {
         foreach (TeleportPad tp in FindObjectsOfType<TeleportPad>())
         {
            if (tp.code == code && tp != this)
            {
                    tp.disableTimer = 2;
                    Vector3 position = tp.gameObject.transform.position;
                    position.y += 2;
                    collider.gameObject.transform.position = position;
                }
            }
        }

    }
}
```

Figure 4.41: Script for teleport.

To give some information about where the teleport will go, create canvas to give information then to make it more interact and nice apply rotation on the canvas.

```
using UnityEngine;
using System.Collections;

public class RotatingObject : MonoBehaviour {


    // Use this for initialization
    void Start ()
    {
 }
    // Update is called once per frame
    void Update ()
    {
        transform.Rotate (new Vector3 (0,Time.deltaTime * 50, 0));
    }
}
```

Figure 4.42: Rotation script for the teleport canvas information.



☑ TeleportGameObject □ Static

Tag Untagged Layer Default

Transform

	X	Y	Z
Position	0	-1.0049	0
Rotation	0	0	0
Scale	0.14050	337.199	0.14050

☑ Rotating Object (Script)

Script RotatingObject

Figure 4.43: Teleport 3D object and the canvas information then apply the script into the canvas.

The hardest part in this scene is to create the interaction script between three different traffic lights. This is because each traffic light must interact with each other. For example if one traffic light is green light then the two traffic lights must be red light, then the traffic light must changing from green light to yellow light then change to red light and so on. The traffic light calculation must be applies into the traffic light 3D object. However, in this scene the traffic light calculation convert the time base. So the light changing only depends to the timer. Table below shows how to apply the logic timing to the three different traffic lights.

	Traffic light 1	Traffic light 2	Traffic light 3
Green	500	500	500
Red	500/150/500/150	500/150/500/150	500/150/500/150
Yellow	150	150	150

Table 4.1: Time waiting for every traffic light

The logic is the green light will start from traffic light 2 then follow by the traffic light 1 then last is traffic light 3 so every light will rotate base on the timing given. Below are the scripts how to apply the script to every traffic light and drag the three different lights to the script.

```
#pragma strict

var RedLightBright : GameObject;
var GreenLightBright : GameObject;
var YellowLightBright : GameObject;

var RedLight: GameObject;
var GreenLight : GameObject;

var RedLightCopy: GameObject;
var GreenLightCopy : GameObject;

var countdownRed : int ;
var countdownYellow : int ;
var countdownGreen : int ;

var redBoolean : boolean ;
var greenBoolean : boolean ;
var yellowBoolean : boolean ;



function Start ()
{
   redBoolean = true;
   greenBoolean = false;
   yellowBoolean = false;

   countdownRed = 650 ;
   countdownGreen = 500 ;
```

```
    countdownYellow = 150 ;

    GreenLightBright.GetComponent.<Renderer>().enabled = false;
    YellowLightBright.GetComponent.<Renderer>().enabled = false;


}

function Update ()
{
    if (redBoolean == true)
    {
        RedLightBright.GetComponent.<Renderer>().enabled = true;
        RedLight.GetComponent.<Renderer>().enabled = false;
        RedLightCopy.GetComponent.<Renderer>().enabled = false;
        countdownRed -= Time.deltaTime;
        if (countdownRed == 0)
        {
            red();
        }
    }
    else if(greenBoolean == true)
    {
        GreenLightBright.GetComponent.<Renderer>().enabled = true;
        GreenLight.GetComponent.<Renderer>().enabled = false;
        GreenLightCopy.GetComponent.<Renderer>().enabled = false;
        countdownGreen -= Time.deltaTime;

        if (countdownGreen == 0)
        {
            green();
        }
    }
```

```
  else if(yellowBoolean == true)
  {
    YellowLightBright.GetComponent.<Renderer>().enabled = true;
    RedLight.GetComponent.<Renderer>().enabled = true;
    GreenLight.GetComponent.<Renderer>().enabled = false;
    RedLightCopy.GetComponent.<Renderer>().enabled = true;
    GreenLightCopy.GetComponent.<Renderer>().enabled = false;
    countdownYellow -= Time.deltaTime;
    if (countdownYellow == 0)
    {
      yellow();
    }
  }
}

function red ()
{
      RedLightBright.GetComponent.<Renderer>().enabled = false;
      RedLight.GetComponent.<Renderer>().enabled = true;
      RedLightCopy.GetComponent.<Renderer>().enabled = true;
      GameObject.Find ("outCubeTraffic1Border1").SendMessage ("moveDown")
;
      GameObject.Find ("outCubeTraffic1Border2").SendMessage ("moveDown")
;
      GameObject.Find ("outCubeTraffic1Border3").SendMessage ("moveDown")
;
      GameObject.Find ("outCubeTraffic1Border4").SendMessage ("moveDown")
;
      redBoolean = false;
      greenBoolean = true;
      countdownRed = 1300;
}
```

```
function green ()
{
	GreenLightBright.GetComponent.<Renderer>().enabled = false;
	GreenLight.GetComponent.<Renderer>().enabled = true;
	GreenLightCopy.GetComponent.<Renderer>().enabled = true;
	GameObject.Find ("outCubeTraffic1Border1").SendMessage ("moveUp");
	GameObject.Find ("outCubeTraffic1Border2").SendMessage ("moveUp");
	GameObject.Find ("outCubeTraffic1Border3").SendMessage ("moveUp");
	GameObject.Find ("outCubeTraffic1Border4").SendMessage ("moveUp");
	greenBoolean = false;
	yellowBoolean = true;
	countdownGreen = 500;
}

function yellow ()
{
	YellowLightBright.GetComponent.<Renderer>().enabled = false;
	RedLight.GetComponent.<Renderer>().enabled = false;
	GreenLight.GetComponent.<Renderer>().enabled = true;
	RedLightCopy.GetComponent.<Renderer>().enabled = false;
	GreenLightCopy.GetComponent.<Renderer>().enabled = true;
	yellowBoolean = false;
	redBoolean = true;
	countdownYellow = 150;
}
```

Figure 4.44: Script that applies into the traffic light 1.

```
#pragma strict


var RedLightBright : GameObject;
var GreenLightBright : GameObject;
var YellowLightBright : GameObject;

var RedLight: GameObject;
var GreenLight : GameObject;

var RedLightCopy: GameObject;
var GreenLightCopy : GameObject;

var countdownRed : int ;
var countdownYellow : int ;
var countdownGreen : int ;

var redBoolean : boolean ;
var greenBoolean : boolean ;
var yellowBoolean : boolean ;

function Start ()
{
   redBoolean = false;
   greenBoolean = true;
   yellowBoolean = false;

   countdownRed = 1300 ;
   countdownGreen = 500 ;
   countdownYellow = 150 ;

   RedLightBright.GetComponent.<Renderer>().enabled = false;
   YellowLightBright.GetComponent.<Renderer>().enabled = false;
}

function Update ()
{
```

```
if (redBoolean == true)
{
    RedLightBright.GetComponent.<Renderer>().enabled = true;
    RedLight.GetComponent.<Renderer>().enabled = false;
    RedLightCopy.GetComponent.<Renderer>().enabled = false;
    countdownRed -= Time.deltaTime;
    if (countdownRed == 0)
    {
        red();
    }
}
else if(greenBoolean == true)
{
    GreenLightBright.GetComponent.<Renderer>().enabled = true;
    GreenLight.GetComponent.<Renderer>().enabled = false;
    GreenLightCopy.GetComponent.<Renderer>().enabled = false;
    countdownGreen -= Time.deltaTime;

    if (countdownGreen == 0)
    {
        green();
    }
}
else if(yellowBoolean == true)
{
    YellowLightBright.GetComponent.<Renderer>().enabled = true;
    RedLight.GetComponent.<Renderer>().enabled = true;
    GreenLight.GetComponent.<Renderer>().enabled = false;
    RedLightCopy.GetComponent.<Renderer>().enabled = true;
    GreenLightCopy.GetComponent.<Renderer>().enabled = false;
    countdownYellow -= Time.deltaTime;
    if (countdownYellow == 0)
```

```
        {
            yellow();
        }
    }
}

function red ()
{
            RedLightBright.GetComponent.<Renderer>().enabled = false;
            RedLight.GetComponent.<Renderer>().enabled = true;
            RedLightCopy.GetComponent.<Renderer>().enabled = true;
            GameObject.Find ("outCubeTraffic2Border1").SendMessage ("moveDown");
            GameObject.Find ("outCubeTraffic2Border2").SendMessage ("moveDown");
            GameObject.Find ("outCubeTraffic2Border3").SendMessage ("moveDown");
            GameObject.Find ("outCubeTraffic2Border4").SendMessage ("moveDown");
            redBoolean = false;
            greenBoolean = true;
            countdownRed = 1300;
}

function green ()
{
            GreenLightBright.GetComponent.<Renderer>().enabled = false;
            GreenLight.GetComponent.<Renderer>().enabled = true;
            GreenLightCopy.GetComponent.<Renderer>().enabled = true;
            GameObject.Find ("outCubeTraffic2Border1").SendMessage ("moveUp");
            GameObject.Find ("outCubeTraffic2Border2").SendMessage ("moveUp");
            GameObject.Find ("outCubeTraffic2Border3").SendMessage ("moveUp");
            GameObject.Find ("outCubeTraffic2Border4").SendMessage ("moveUp");
            greenBoolean = false;
            yellowBoolean = true;
            countdownGreen = 500;
}

function yellow ()
{
            YellowLightBright.GetComponent.<Renderer>().enabled = false;
            RedLight.GetComponent.<Renderer>().enabled = false;
            GreenLight.GetComponent.<Renderer>().enabled = true;
            RedLightCopy.GetComponent.<Renderer>().enabled = false;
            GreenLightCopy.GetComponent.<Renderer>().enabled = true;
            yellowBoolean = false;
            redBoolean = true;
            countdownYellow = 150;
}
```

Figure 4.45: Script that applies into the traffic light 2.

```
#pragma strict

var RedLightBright : GameObject;
var GreenLightBright : GameObject;
var YellowLightBright : GameObject;

var RedLight: GameObject;
var GreenLight : GameObject;

var RedLightCopy: GameObject;
var GreenLightCopy : GameObject;

var countdownRed : int ;
var countdownYellow : int ;
var countdownGreen : int ;

var redBoolean : boolean ;
var greenBoolean : boolean ;
var yellowBoolean : boolean ;

function Start ()
{
    redBoolean = true;
    greenBoolean = false;
    yellowBoolean = false;

    countdownRed = 1300 ;
    countdownGreen = 500 ;
    countdownYellow = 150 ;

    GreenLightBright.GetComponent.<Renderer>().enabled = false;
    YellowLightBright.GetComponent.<Renderer>().enabled = false;
}

function Update ()
{
    if (redBoolean == true)
    {
        RedLightBright.GetComponent.<Renderer>().enabled = true;
        RedLight.GetComponent.<Renderer>().enabled = false;
        RedLightCopy.GetComponent.<Renderer>().enabled = false;
        countdownRed -= Time.deltaTime;
        if (countdownRed == 0)
        {
            red();
        }
    }
    else if(greenBoolean == true)
    {
        GreenLightBright.GetComponent.<Renderer>().enabled = true;
        GreenLight.GetComponent.<Renderer>().enabled = false;
        GreenLightCopy.GetComponent.<Renderer>().enabled = false;
        countdownGreen -= Time.deltaTime;

        if (countdownGreen == 0)
        {
            green();
        }
    }
    else if(yellowBoolean == true)
    {
        YellowLightBright.GetComponent.<Renderer>().enabled = true;
        RedLight.GetComponent.<Renderer>().enabled = true;
        GreenLight.GetComponent.<Renderer>().enabled = false;
        RedLightCopy.GetComponent.<Renderer>().enabled = true;
        GreenLightCopy.GetComponent.<Renderer>().enabled = false;
        countdownYellow -= Time.deltaTime;
        if (countdownYellow == 0)
        {
            yellow();
        }
    }
}

function red ()
{
        RedLightBright.GetComponent.<Renderer>().enabled = false;
        RedLight.GetComponent.<Renderer>().enabled = true;
        RedLightCopy.GetComponent.<Renderer>().enabled = true;
        GameObject.Find ("outCubeTraffic3Border1").SendMessage ("moveDown")
;
        GameObject.Find ("outCubeTraffic3Border2").SendMessage ("moveDown")
;
        GameObject.Find ("outCubeTraffic3Border3").SendMessage ("moveDown")
;
        GameObject.Find ("outCubeTraffic3Border4").SendMessage ("moveDown")
;
        redBoolean = false;
        greenBoolean = true;
        countdownRed = 1300;
}

function green ()
{
        GreenLightBright.GetComponent.<Renderer>().enabled = false;
        GreenLight.GetComponent.<Renderer>().enabled = true;
        GreenLightCopy.GetComponent.<Renderer>().enabled = true;
        GameObject.Find ("outCubeTraffic3Border1").SendMessage ("moveUp");
        GameObject.Find ("outCubeTraffic3Border2").SendMessage ("moveUp");
        GameObject.Find ("outCubeTraffic3Border3").SendMessage ("moveUp");
        GameObject.Find ("outCubeTraffic3Border4").SendMessage ("moveUp");
        greenBoolean = false;
        yellowBoolean = true;
        countdownGreen = 500;
}

function yellow ()
{
        YellowLightBright.GetComponent.<Renderer>().enabled = false;
        RedLight.GetComponent.<Renderer>().enabled = false;
        GreenLight.GetComponent.<Renderer>().enabled = true;
        RedLightCopy.GetComponent.<Renderer>().enabled = false;
        GreenLightCopy.GetComponent.<Renderer>().enabled = true;
        yellowBoolean = false;
        redBoolean = true;
        countdownYellow = 150;
}
```

Figure 4.46: Script that applies into the traffic light 3.

After attach all the script to the right number of traffic light. Drag and drop the traffic light colour, which are red, green and yellow to the slot the created in the script. The process drag and drop almost same to the all types but the different is what 3D object that need to drag and drop to the script.

To control the movement or give instruction to the user in this scene, apply border to every road. The main function this border is to make sure the user does not cross the road without using the zebra crossing. The script needs to apply like below.

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;

public class RoadBorderScript : MonoBehaviour {

   public Text leftText;
   public Text rightText;
   public float fadeSpeed = 5f;
   public bool entrance;

   void Start ()
   {
       leftText.text = "";
       rightText.text = "";
   }

   void OnTriggerEnter(Collider collider)
   {
       entrance = true;
       if (collider.gameObject.name == "FPSController")
       {
           leftText.text = "Do not Cross Here.\nUse Zebra Crossing.";
           rightText.text = "Do not Cross Here.\nUse Zebra Crossing.";
       }

   }

   void OnTriggerExit (Collider collider) {
       entrance = false;
       leftText.text = "";
       rightText.text = "";

   }
}
```

Figure 4.47: Script that applies into the road border.

```
using UnityEngine;
using UnityEngine.SceneManagement;
using System.Collections;

public class CarradgeOneMoving : MonoBehaviour {

    float speed = 20.0f;
    int countdownRed;
```

```
bool someBoolean;
// Use this for initialization
void Start ()
{
    someBoolean = false;
    countdownRed = 652;

}

// Update is called once per frame
void Update ()
{
    moving();
}

void OnCollisionEnter(Collision collision)
{
    if(collision.gameObject.name == "FPSController")
    {
        //Destroy(collision.gameObject);
        SceneManager.LoadScene (2);
        //Application.LoadLevel(Application.loadedLevel);
    }

    else if(collision.gameObject.name == "DestroyCar")
    {
        Destroy(this.gameObject);
    }
}

void moving()
{
        if (someBoolean)
        {

            transform.Translate (Vector3.forward * speed * Time.deltaTime);

        }

        else if (!someBoolean)
        {
            countdownRed -= 1;
            if (countdownRed == 0)
            {
                someBoolean = true;
                GameObject.Find ("CarrageOneWheelRR").SendMessage ("ContRotate");
                GameObject.Find ("CarrageOneWheelRL").SendMessage ("ContRotate");
                GameObject.Find ("CarrageOneWheelFL").SendMessage ("ContRotate");
                GameObject.Find ("CarrageOneWheelFR").SendMessage ("ContRotate");
            }
        }
    }
}
```

Figure 4.48: Script that applies into the CarrageOne

```
using UnityEngine;
using System.Collections;

public class CarradgeOneWheel : MonoBehaviour {

    float speed = 600.0f;
    bool someBoolean;
    // Use this for initialization
    void Start ()
    {
        someBoolean = false;
    }

    // Update is called once per frame
    void Update ()
    {
        moving();
    }
    void ContRotate()
    {
        someBoolean = true;
    }

    void moving()
    {
        if (someBoolean)
        {
            transform.Rotate (new Vector3 (speed * Time.deltaTime,0, 0));
        }
    }
}
```

Figure 4.49: Script that applies into the CarrageOneWheel

```
using UnityEngine;
using UnityEngine.SceneManagement;
using System.Collections;

public class CarradgeThreeMoving : MonoBehaviour {

    float speed = 20.0f;
    int countdownRed ;
```

```
bool someBoolean;

// Use this for initialization
void Start ()
{
    someBoolean = true;
    //countdownRed = 1 ;
}

// Update is called once per frame
void Update ()
{
    moving();

    //if (GameObject.Find ("FPSController") == null)
    //{
    //    Application.LoadLevel(Application.loadedLevel);
    //}
}

void OnTriggerEnter(Collider collider)
{
    if (collider.gameObject.name == "Stop")
    {
        someBoolean = false;
        countdownRed = 250;
        GameObject.Find ("CarrageThreeWheelRL").SendMessage ("StopRotate");
        GameObject.Find ("CarrageThreeWheelRR").SendMessage ("StopRotate");
        GameObject.Find ("CarrageThreeWheelFL").SendMessage ("StopRotate");
        GameObject.Find ("CarrageThreeWheelFR").SendMessage ("StopRotate");
    }
```

```
}

void OnCollisionEnter(Collision collision)
{
    if(collision.gameObject.name == "FPSController")
    {
        //Destroy(collision.gameObject);
        SceneManager.LoadScene (2);
    }

    else if(collision.gameObject.name == "DestroyCar")
    {
        Destroy(this.gameObject);
    }
}

void moving()
{
    if (someBoolean)
    {

        transform.Translate (Vector3.forward * speed * Time.deltaTime);

    }

    else if (!someBoolean)
    {
        countdownRed -= 1;
        if (countdownRed == 0)
        {
            someBoolean = true;
            GameObject.Find ("CarrageThreeWheelRL").SendMessage ("ContRotate")
;
            GameObject.Find ("CarrageThreeWheelRR").SendMessage ("ContRotate")
;
            GameObject.Find ("CarrageThreeWheelFL").SendMessage ("ContRotate");
            GameObject.Find ("CarrageThreeWheelFR").SendMessage ("ContRotate")
;
          }
        }
    }
}
```

Figure 4.50: Script that applies into the CarrageThree

```
using UnityEngine;
using UnityEngine.SceneManagement;
using System.Collections;

public class CarradgeTwoMoving : MonoBehaviour {

    float speed = 20.0f;
    int countdownRed;
    bool someBoolean;



    // Use this for initialization
    void Start ()
    {
        someBoolean = true;
        //countdownRed = 1 ;

    }

    // Update is called once per frame
```

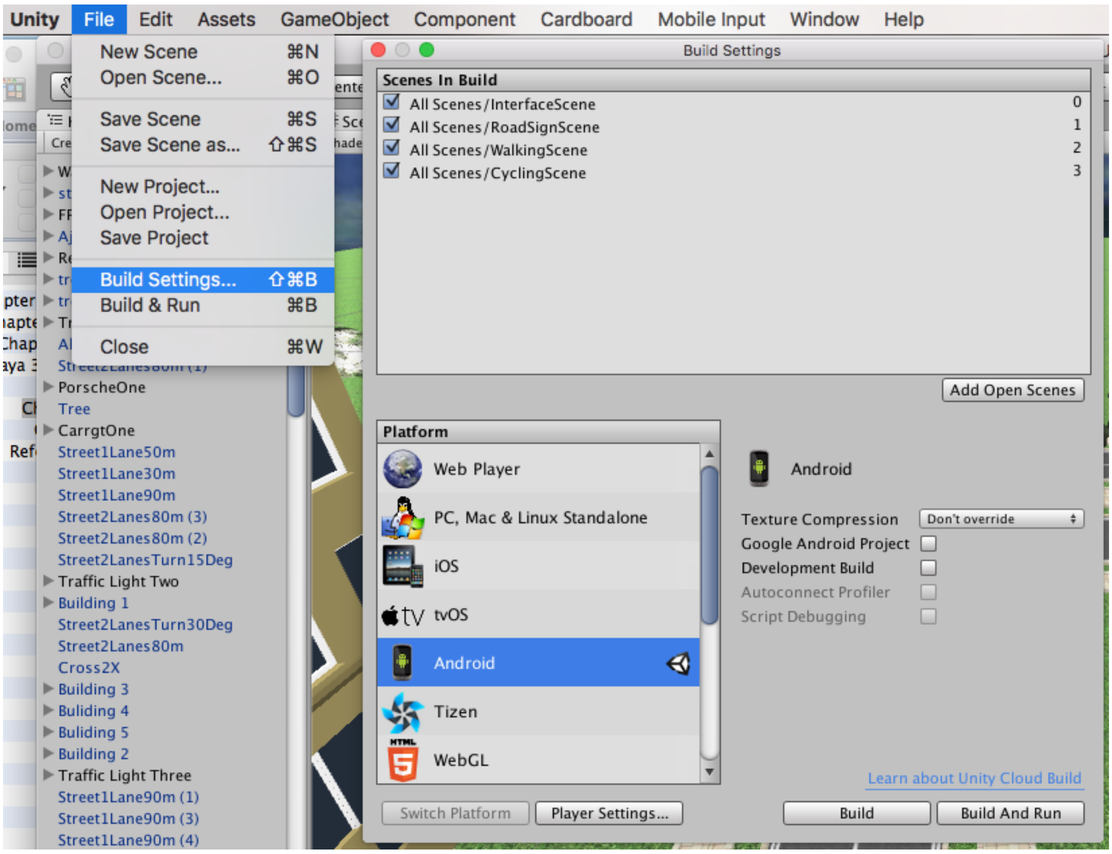
```
void Update ()
{
    moving();
}

void OnCollisionEnter(Collision collision)
{
    if(collision.gameObject.name == "FPSController")
    {
        //Destroy(collision.gameObject);
        SceneManager.LoadScene (2);
    }

    else if(collision.gameObject.name == "DestroyCarTwo")
    {
        Destroy(this.gameObject);
    }
}

void OnTriggerEnter(Collider collider)
{
    if (collider.gameObject.name == "StopTwo")
    {
        someBoolean = false;
        countdownRed = 500;
        GameObject.Find ("CarrageTwoWheelRL").SendMessage ("StopRotate");
        GameObject.Find ("CarrageTwoWheelRR").SendMessage ("StopRotate");
        GameObject.Find ("CarrageTwoWheelFL").SendMessage ("StopRotate");
        GameObject.Find ("CarrageTwoWheelFR").SendMessage ("StopRotate");
    }

}
```

```
    void moving()
    {
        if (someBoolean)
        {

            transform.Translate (Vector3.forward * speed * Time.deltaTime);

        }

        else if (!someBoolean)
        {
            countdownRed -= 1;
            if (countdownRed == 0)
            {
                someBoolean = true;
                GameObject.Find ("CarrageTwoWheelRL").SendMessage ("ContRotate");
                GameObject.Find ("CarrageTwoWheelRR").SendMessage ("ContRotate");
                GameObject.Find ("CarrageTwoWheelFL").SendMessage ("ContRotate");
                GameObject.Find ("CarrageTwoWheelFR").SendMessage ("ContRotate");
            }
        }
    }
}
```

Figure 4.51: Script that applies into the CarrageTwo

4.4 BUILD THE PROJECT

To build this project go to File > Build Setting > Android > Build. Before build in android platform make sure Unity has the Android SDK. To build in iOS platform just hit the button switch platform.



Scenes In Build

☑ All Scenes/InterfaceScene 0
☑ All Scenes/RoadSignScene 1
☑ All Scenes/WalkingScene 2
☑ All Scenes/CyclingScene 3

Android

Texture Compression: Don't override
Google Android Project □
Development Build □
Autoconnect Profiler □
Script Debugging □

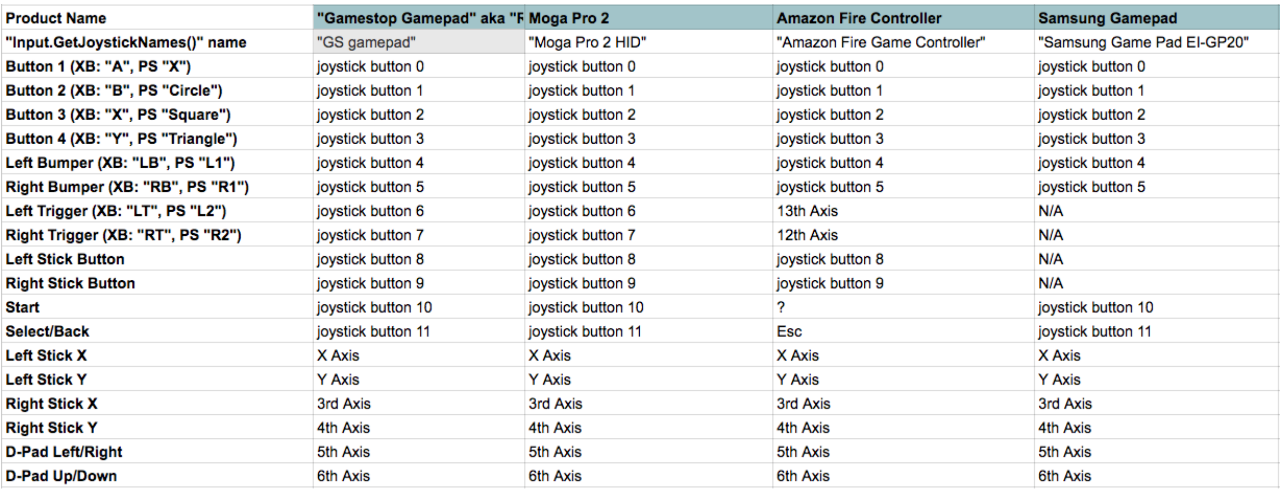
Figure 4.52: Build to android platform

4.5 INSTALL TO THE SMARTPHONE

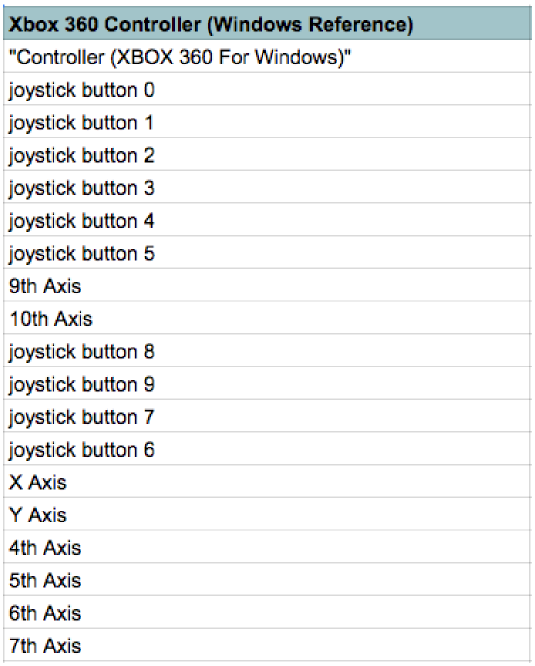
After build the project the output must be in the .apk format. Put the .apk file into the smartphone then installed it.

4.6 GAMEPAD CONTROLLER BUTTON

Below is example button and the different type of controller will work with unity.



Product Name	"Gamestop Gamepad" aka "F	Moga Pro 2	Amazon Fire Controller	Samsung Gamepad
"Input.GetJoystickNames()" name	"GS gamepad"	"Moga Pro 2 HID"	"Amazon Fire Game Controller"	"Samsung Game Pad EI-GP20"
Button 1 (XB: "A", PS "X")	joystick button 0	joystick button 0	joystick button 0	joystick button 0
Button 2 (XB: "B", PS "Circle")	joystick button 1	joystick button 1	joystick button 1	joystick button 1
Button 3 (XB: "X", PS "Square")	joystick button 2	joystick button 2	joystick button 2	joystick button 2
Button 4 (XB: "Y", PS "Triangle")	joystick button 3	joystick button 3	joystick button 3	joystick button 3
Left Bumper (XB: "LB", PS "L1")	joystick button 4	joystick button 4	joystick button 4	joystick button 4
Right Bumper (XB: "RB", PS "R1")	joystick button 5	joystick button 5	joystick button 5	joystick button 5
Left Trigger (XB: "LT", PS "L2")	joystick button 6	joystick button 6	13th Axis	N/A
Right Trigger (XB: "RT", PS "R2")	joystick button 7	joystick button 7	12th Axis	N/A
Left Stick Button	joystick button 8	joystick button 8	joystick button 8	N/A
Right Stick Button	joystick button 9	joystick button 9	joystick button 9	N/A
Start	joystick button 10	joystick button 10	?	joystick button 10
Select/Back	joystick button 11	joystick button 11	Esc	joystick button 11
Left Stick X	X Axis	X Axis	X Axis	X Axis
Left Stick Y	Y Axis	Y Axis	Y Axis	Y Axis
Right Stick X	3rd Axis	3rd Axis	3rd Axis	3rd Axis
Right Stick Y	4th Axis	4th Axis	4th Axis	4th Axis
D-Pad Left/Right	5th Axis	5th Axis	5th Axis	5th Axis
D-Pad Up/Down	6th Axis	6th Axis	6th Axis	6th Axis



Xbox 360 Controller (Windows Reference)
"Controller (XBOX 360 For Windows)"
joystick button 0
joystick button 1
joystick button 2
joystick button 3
joystick button 4
joystick button 5
9th Axis
10th Axis
joystick button 8
joystick button 9
joystick button 7
joystick button 6
X Axis
Y Axis
4th Axis
5th Axis
6th Axis
7th Axis

Table 4.2: Button Controller (Source: NurFACEGAMES)

4.7 TESTING ON END USERS

On this Phase, the system functionality must be test based on User Acceptance Test (UAT). Approval of this testing implies that reviews ore confident that following the execution of the test plan, the resulting system will be considered fully-tested and eligible for implementation. The chosen user will go through each of the instruction in the user manual. Any error or problems found by the user must be noted on the form.

In this case, the users that tested this system they are from SK. Gambang and also the teacher at that school. The class that chosen to test this system is class 5 Fatonah. 30 participants has been involved in this testing. Below is the result of UAT.

Event	**Test Data**	**Expected Result**	**Actual Result**	**Pass/ Fail**	**Comment**
Exit	Exit_Button	Quit the system	System quit if button selected	PASS	
Road Sign	RoadSign_Button	Go to scene One	Enter the scene One	PASS	
Zebra Crossing	ZebraCrossing_ Button	Go to scene Two	Enter the scene One	PASS	
Menu Scene One	Menu_Button	Exit and go to main menu	Exit and go to main menu	PASS	
Menu Scene Two	Menu_Button	Exit and go to main menu	Exit and go to main menu	PASS	

Table 4.3: Test Case Design

Test case name:	Check response for application installation			**Reference:**	SDD/SRS
Test case version:	TC_Ver1	**Sheet:**	1	**Date created:**	16/05/2016

Task or requirement	**Ref.** *incl page no.*	**Fail** *issue no.*	**Query** *issue No.*	**Pass**
i) The application is installed completely inside the smartphone. It response smoothly without stocked or hanged when the application is selected.	-	-	-	Pass

Test case name:	Check response for Menu Scene functionality			**Reference:**	SDD/SRS
Test case version:	TC_Ver1	**Sheet:**	2	**Date created:**	16/05/2016

Task or requirement	**Ref.** *incl page no.*	**Fail** *issue no.*	**Query** *issue No.*	**Pass**
i) The image Button animated well and displayed the all three buttons.	-	-	-	Pass

Test case name:	Check response for Road Signs in Menu Scene functionality			**Reference:**	SDD/SRS
Test case version:	TC_Ver1	**Sheet:**	3	**Date created:**	16/05/2016

Task or requirement	**Ref.** *incl page no.*	**Fail** *issue no.*	**Query** *issue No.*	**Pass**
i) “Road Sings” button response to the “Road Signs” Scene.	-	-	-	Pass

Test case name:	Check response for Zebra Crossing in Menu Scene functionality			**Reference:**	SDD/SRS
Test case version:	TC_Ver1	**Sheet:**	4	**Date created:**	16/05/2016

Task or requirement		**Ref.** *incl page no.*	**Fail** *issue no.*	**Query** *issue No.*	**Pass**
i)	“Zebra Crossing” button response to the “Zebra Crossing” Scene.	-	-	-	Pass

Test case name:	Check response for Menu in Road signs Scene functionality			**Reference:**	SDD/SRS
Test case version:	TC_Ver1	**Sheet:**	5	**Date created:**	16/05/2016

Task or requirement		**Ref.** *incl page no.*	**Fail** *issue no.*	**Query** *issue No.*	**Pass**
i)	“Menu” button response to the “Menu Scene” Scene.	-	-	-	Pass

Test case name:	Check response for Menu in Zebra Crossing Scene functionality			**Reference:**	SDD/SRS
Test case version:	TC_Ver1	**Sheet:**	6	**Date created:**	16/05/2016

Task or requirement		**Ref.** *incl page no.*	**Fail** *issue no.*	**Query** *issue No.*	**Pass**
i)	“Menu” button response to the “Menu Scene” Scene.	-	-	-	Pass

Table 4.4: User Acceptance Test Checklist

4.8 RESULT ON OBSERVATION

After made the end user testing for the application functionality such us button and the walkthrough. The observation analysis has been made to support the objective this project. Below is the result of the observations analysis.

4.8.1 APPLICATION CONTENT

For this analysis, users have been asked about the contents contained in this application. This type of analysis only focusing on the content on this application for example, the first scene is about understanding the meaning of every road sign. The example question that asked is:

i. Did you know about road safety, Yes / No?
ii. How you learn about road safety, Book / Web site / 3D animation?
iii. Did you know every meaning of the road signs, Yes / No?
iv. Did you know every meaning of the color for the road signs, Yes / No?

These four questions asked to the user before using and testing this application. The result for every question has been stated below.

Student No.	Question 1	Question 2	Question 3	Question 4
1	No	Book	No	No
2	No	Book	No	No
3	Yes	Book	Yes	Yes
4	No	Web Site	No	No
5	No	Book	No	No
6	No	Book	No	No

7	No	Book	No	No
8	Yes	Book	Yes	Yes
9	No	Book	No	No
10	No	Web Site	No	Yes
11	Yes	Web Site	Yes	Yes
12	No	Book	No	No
13	No	Book	No	No
14	Yes	Book	Yes	No
15	No	Web Site	No	Yes
16	Yes	Web Site	Yes	No
17	No	Book	No	No
18	Yes	Book	Yes	No
19	Yes	Web Site	Yes	No
20	Yes	Book	Yes	No
21	No	Book	No	No
22	Yes	Book	Yes	No
23	Yes	Book	Yes	No
24	Yes	Web Site	Yes	Yes
25	No	Book	No	No
26	Yes	Book	Yes	Yes
27	No	Web Site	No	No
28	Yes	Book	Yes	No
29	No	Book	No	No
30	Yes	Book	Yes	No

Table 4.5: Observation resulted before testing

From this observations and analysis it can be conclude that most of the student did not know about the road safety, meaning of the road signs and the meaning of the road sign’s color. Focusing on the question two, most of the students learn about the road safety from book. Thus, it shows that students today learned the road safety knowledge from theoretical only.

So, from this propose application it can be help the user to give the understanding about the road safety then not only focusing on theoretical from book but also give immersive learning and interaction in learning process.

After make and taken the result. The users start to test the application and run the entire scene that has in the application. After testing and run the application another analysis has been taken. The purpose of the second analysis is to observe the user reaction after using the application.

Based on the observation, the user have shown their interest in demonstrated the system for using the application. The best part is the user had demonstrated their enthusiasm and extremely energized in utilizing this application. It can be seen on their outward appearance when each one of them tried the application. In light of the observation, it has been demonstrated that this application can be a device that draw in enthusiasm from the user to learn road safety.

Below is the result of the observation from 30 students after using the application. Most of them show the interested, exploring and enjoyable after using the application.

Student No.	**Interested**	**Exploring**	**Enjoyable**
1	Yes	Yes	Yes
2	Yes	No	Yes
3	Yes	Yes	No
4	Yes	Yes	Yes
5	Yes	Yes	No
6	Yes	Yes	Yes
7	Yes	No	Yes
8	Yes	No	No
9	Yes	Yes	Yes
10	Yes	Yes	Yes
11	Yes	Yes	Yes
12	Yes	Yes	Yes

13	Yes	No	No
14	Yes	Yes	Yes
15	Yes	Yes	No
16	Yes	Yes	Yes
17	Yes	Yes	Yes
18	Yes	Yes	Yes
19	Yes	Yes	Yes
20	Yes	Yes	No
21	Yes	Yes	Yes
22	Yes	No	No
23	Yes	Yes	Yes
24	Yes	No	Yes
25	Yes	Yes	Yes
26	Yes	No	Yes
27	Yes	Yes	No
28	Yes	Yes	No
29	Yes	Yes	Yes
30	Yes	Yes	Yes

Table 4.6: Observation resulted after testing

From this second observation, it can be seen that this application help user in learning process to more interactive and knowledgeable. These propose application also give enjoyable rather than only read the theoretical book. The user also interest to test the application longer than the time giving. The users gave 3 minute to test this application. The time giving brings to the second observation about the simulation sickness.

4.8.2 OBSERVATION OF SIMULATION SICKNESS

The purpose of this observation is to measure the simulation sickness by time giving after using the VR box. A different person has their own level of simulation sickness. In this observation, the way to measure the simulation sickness is from the time by the different age. The user that tested this application is from Univeristi Malaysia Pahang. Below is the output from the observation.

Age (Year Old)	Time taken(Min)
10-12	8-10
22-24	4-6

Table 4.7: Simulation sickness result

From the result above, the younger user take a longer timer then the older user to have the simulation sickness. However, not all the user that has tested the system will might the simulation sickness. Thus, this observation can be conclude that, the simulation sickness do not depends on the people age but it occur because of naturally has for every people. Only certain people might have this problem. To them who have felt that experience they need to rest and drink water plain water.

CHAPTER 5

CONCLUSION

5.1 INTRODUCTION

From this system development, the objectives and scopes are identified as the way to solve the problem statement that has made. It is also contains explanation in order to facilities the benefits that can be gathered throughout the system for children. By implementing this system, in future with hope it can overcome the current problems and gives the advantages to the children to learn about road safety. In fulfilling the objectives in development process of this system, RAD methodology had been used. This approach help in running framework grow speedier and take after the progression. The development was separated into four phases. There are Requirements planning phase, User design phase, Construction phase, and cutover phase. In construction phase the script C# and JavaScript had been used to write the code in this system.

5.2 LIMITATION

Behind this project, there have imperatives in some of procedure. To begin with of all, the Autodesk Maya software for making 3D object model. This software needs the high specification of computer to run it. Beside that this software also need a licence to run it as professional software. After that, another issue come is using Unity 3D software. This software is free, however in free software the user cannot develop more than three scene. Because of that, this project only has three scenes although for the first it has four scenes.

To fulfil all the client needed and approve the user design and also the requirement. It needs to have a discussion with the client. It is difficult to meet the client because of the work schedule. Thus, only few meeting can be done with the client.

In term of application, users need to have the VR box and the controller to run and communicate with the application scene. By that, it will complete all the stages created inside the application.

5.3 ADVANTAGES AND DISADVANTAGES

From this application, there are points of interest and weaknesses. This application shows the point of view in 360 degree of view. Using the VR box the user can immerse to the scene in the application. User can view the environment by look around and turn around their head. Besides that, this application also imbedded with music for every scene. The case of interactivity, this system has full interactivity. The user can interact using the controller to walk, run, jump and other in this system. By using this application the user can get knowledge about road safety.

Disadvantage of this application is it needs the high end smartphone to run this application. Without high smart phone the application cannot be run. Besides that, this application also needs a large of space installation. This application uses a real scale of 3D object in development. So this application, need a large of space to store in the smart phone.

5.4 FUTURE WORK

This system needs to add more module and rule to know the period of their learning, especially in this field, which is in road safety learning tools. In this system just have two scenes that teach about road signs and how to cross with zebra crossing. This system supposes has more than two scenes to make it more knowledgeable. The license of unity 3D must be grabbed to exploring this VR using unity.

Another territory that can be enhances, this system not only for the android platforms but also in iOS platforms. This is because; these two types of operating system are most popular in this country. To make this system can be access for any type of smart phones it must use able in any operating system.

REFERENCES

D. Waller, E. Bachmann, E. Hodgson, A.C. Beall (2007). *The HIVE: a huge immersive virtual environment for research in spatial cognition*. Behavior Research Methods, 39, 835 – 843.

P.W. Fink, P.S. Foo, W.H. Warren (2009). *Catching fly balls in virtual reality: a critical test of the outfielder problem.* Journal of Vision, 9, 1 – 8.

James D. Wright Christopher R. Wolfe, Elizabeth M. Cedillos (2015). *International Encyclopedia of the Social & Behavioral Sciences (Second Edition).* E-Communications Platforms and E-Learning, 7, 895 – 902.

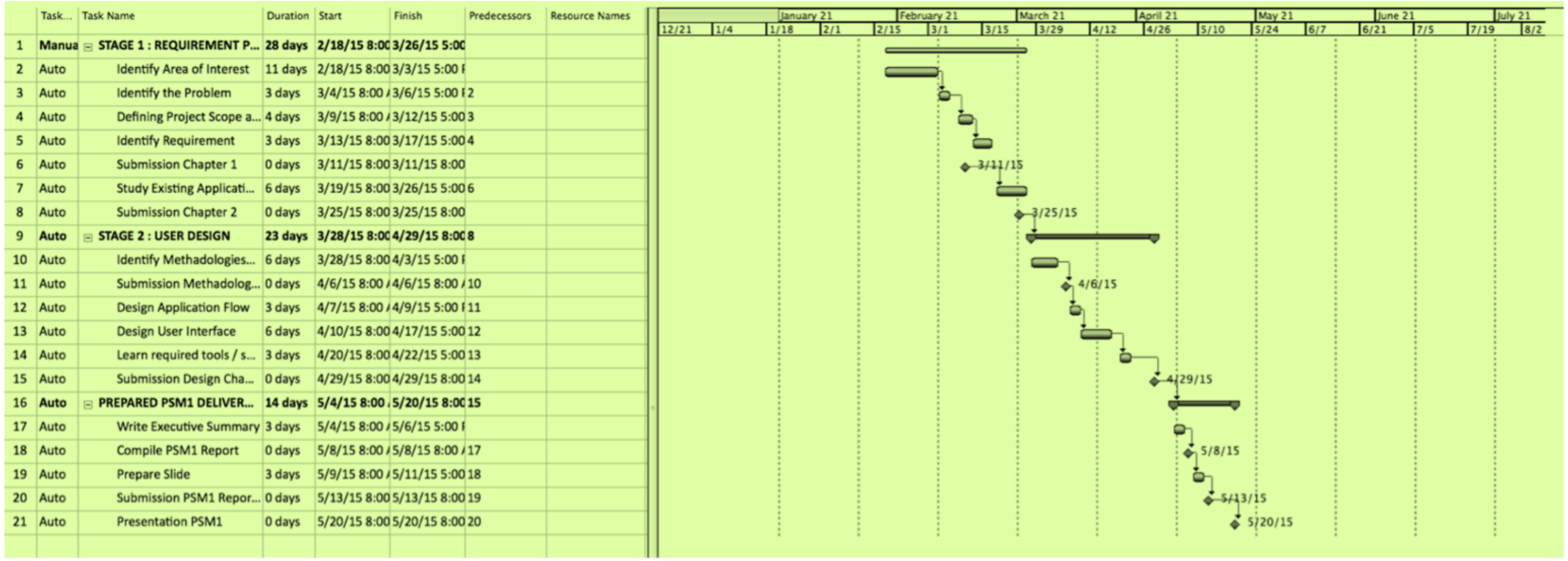
Peden, M., Scurfield, R., Sleet, D., Mohan, D., Hyder, A. A., Jarawan, E., Mathers, C (2004). *Global status report on road safety 2013: Violence and Injury Prevention*. World report on road traffic injury prevention, 167 – 197.

Al Haji, G (2005). *Towards a Road Safety Development Index (RSDI)*. Development of an International Index to Measure Road Safety Performance. Linköping University, Norrköping, Sweden.

Hu, Fei (2016). *Virtual Reality Enhanced Robotic Systems for Disability Rehabilitation.* Gallery. 49. ISBN 978-1466697409.

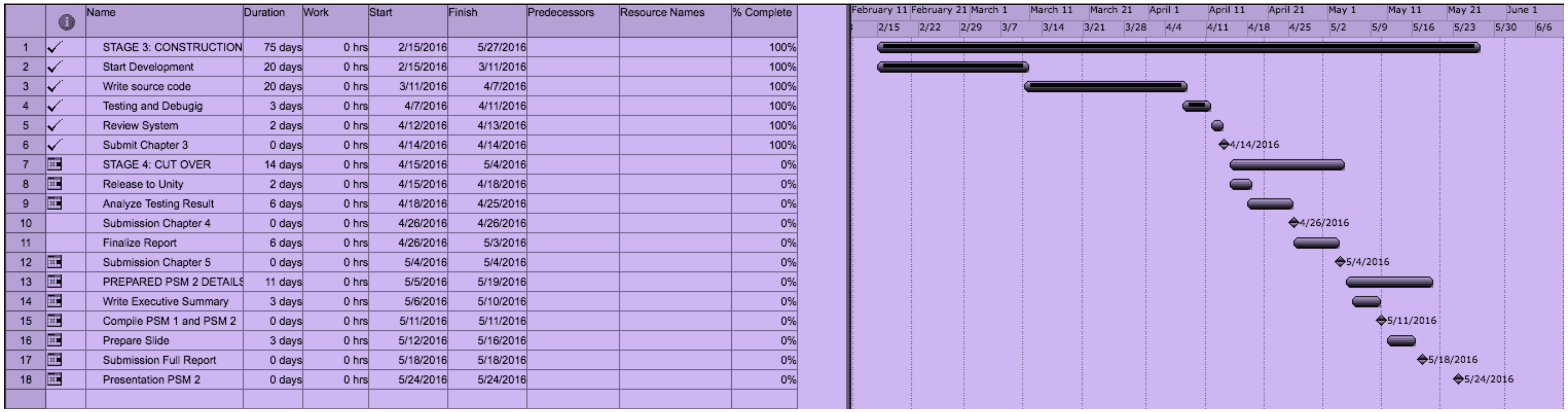
APPENDIX

Gantt chart PSM I



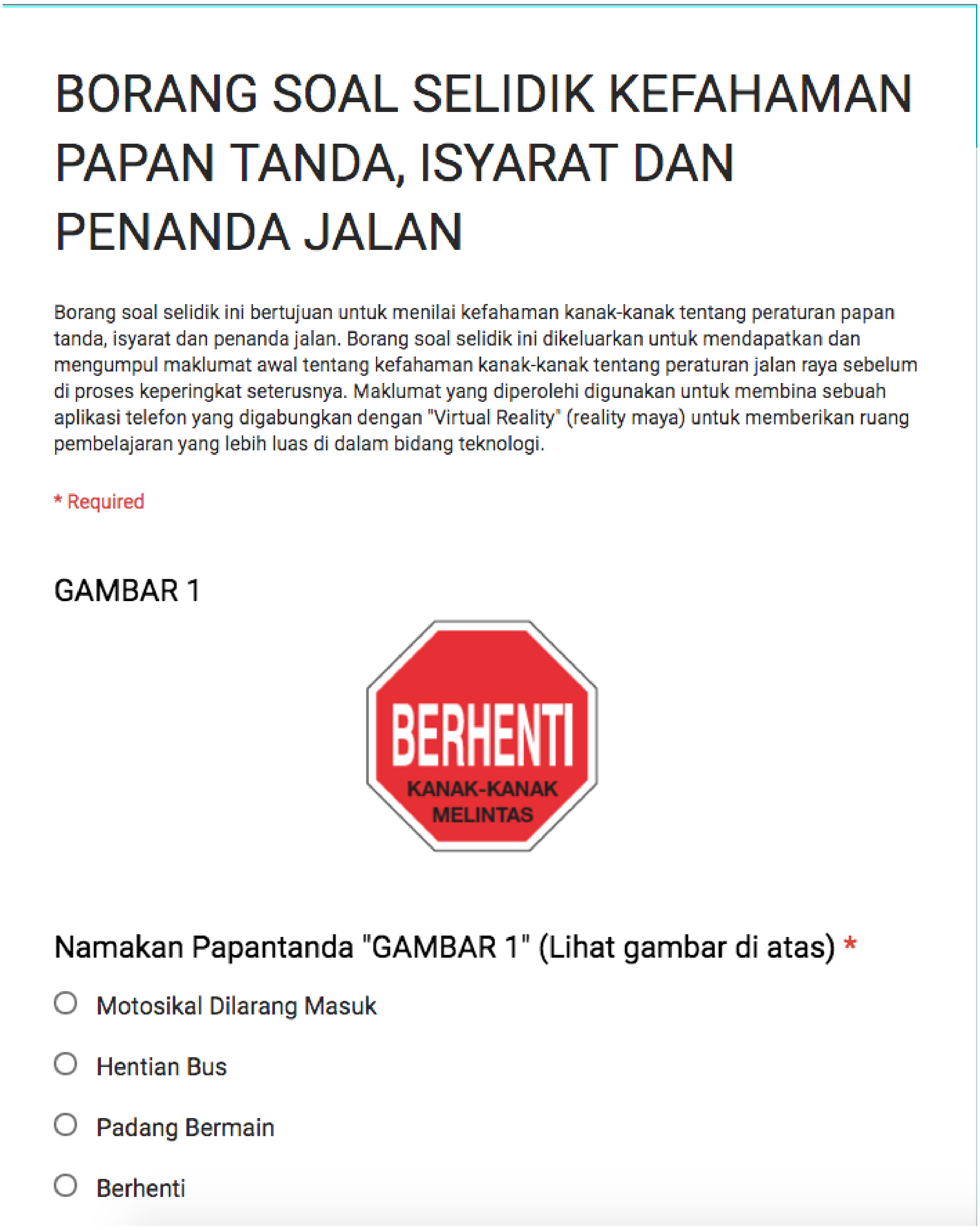
	Task...	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	**Manua**	⊟ **STAGE 1 : REQUIREMENT P...**	**28 days**	**2/18/15 8:00**	**3/26/15 5:00**		
2	Auto	Identify Area of Interest	11 days	2/18/15 8:00	3/3/15 5:00 F		
3	Auto	Identify the Problem	3 days	3/4/15 8:00 /	3/6/15 5:00 F	2	
4	Auto	Defining Project Scope a...	4 days	3/9/15 8:00 /	3/12/15 5:00	3	
5	Auto	Identify Requirement	3 days	3/13/15 8:00	3/17/15 5:00	4	
6	Auto	Submission Chapter 1	0 days	3/11/15 8:00	3/11/15 8:00		
7	Auto	Study Existing Applicati...	6 days	3/19/15 8:00	3/26/15 5:00	6	
8	Auto	Submission Chapter 2	0 days	3/25/15 8:00	3/25/15 8:00		
9	**Auto**	⊟ **STAGE 2 : USER DESIGN**	**23 days**	**3/28/15 8:00**	**4/29/15 8:00**	**8**	
10	Auto	Identify Methadologies...	6 days	3/28/15 8:00	4/3/15 5:00 F		
11	Auto	Submission Methadolog...	0 days	4/6/15 8:00 /	4/6/15 8:00 /	10	
12	Auto	Design Application Flow	3 days	4/7/15 8:00 /	4/9/15 5:00 F	11	
13	Auto	Design User Interface	6 days	4/10/15 8:00	4/17/15 5:00	12	
14	Auto	Learn required tools / s...	3 days	4/20/15 8:00	4/22/15 5:00	13	
15	Auto	Submission Design Cha...	0 days	4/29/15 8:00	4/29/15 8:00	14	
16	**Auto**	⊟ **PREPARED PSM1 DELIVER...**	**14 days**	**5/4/15 8:00 /**	**5/20/15 8:00**	**15**	
17	Auto	Write Executive Summary	3 days	5/4/15 8:00 /	5/6/15 5:00 F		
18	Auto	Compile PSM1 Report	0 days	5/8/15 8:00 /	5/8/15 8:00 /	17	
19	Auto	Prepare Slide	3 days	5/9/15 8:00 /	5/11/15 5:00	18	
20	Auto	Submission PSM1 Repor...	0 days	5/13/15 8:00	5/13/15 8:00	19	
21	Auto	Presentation PSM1	0 days	5/20/15 8:00	5/20/15 8:00	20	

Gantt chart PSM II



		Name	Duration	Work	Start	Finish	Predecessors	Resource Names	% Complete
1	✓	STAGE 3: CONSTRUCTION	75 days	0 hrs	2/15/2016	5/27/2016			100%
2	✓	Start Development	20 days	0 hrs	2/15/2016	3/11/2016			100%
3	✓	Write source code	20 days	0 hrs	3/11/2016	4/7/2016			100%
4	✓	Testing and Debugig	3 days	0 hrs	4/7/2016	4/11/2016			100%
5	✓	Review System	2 days	0 hrs	4/12/2016	4/13/2016			100%
6	✓	Submit Chapter 3	0 days	0 hrs	4/14/2016	4/14/2016			100%
7	▦	STAGE 4: CUT OVER	14 days	0 hrs	4/15/2016	5/4/2016			0%
8	▦	Release to Unity	2 days	0 hrs	4/15/2016	4/18/2016			0%
9	▦	Analyze Testing Result	6 days	0 hrs	4/18/2016	4/25/2016			0%
10		Submission Chapter 4	0 days	0 hrs	4/26/2016	4/26/2016			0%
11		Finalize Report	6 days	0 hrs	4/26/2016	5/3/2016			0%
12	▦	Submission Chapter 5	0 days	0 hrs	5/4/2016	5/4/2016			0%
13	▦	PREPARED PSM 2 DETAILS	11 days	0 hrs	5/5/2016	5/19/2016			0%
14	▦	Write Executive Summary	3 days	0 hrs	5/6/2016	5/10/2016			0%
15	▦	Compile PSM 1 and PSM 2	0 days	0 hrs	5/11/2016	5/11/2016			0%
16	▦	Prepare Slide	3 days	0 hrs	5/12/2016	5/16/2016			0%
17	▦	Submission Full Report	0 days	0 hrs	5/18/2016	5/18/2016			0%
18	▦	Presentation PSM 2	0 days	0 hrs	5/24/2016	5/24/2016			0%

Questionnaire



BORANG SOAL SELIDIK KEFAHAMAN PAPAN TANDA, ISYARAT DAN PENANDA JALAN

Borang soal selidik ini bertujuan untuk menilai kefahaman kanak-kanak tentang peraturan papan tanda, isyarat dan penanda jalan. Borang soal selidik ini dikeluarkan untuk mendapatkan dan mengumpul maklumat awal tentang kefahaman kanak-kanak tentang peraturan jalan raya sebelum di proses keperingkat seterusnya. Maklumat yang diperolehi digunakan untuk membina sebuah aplikasi telefon yang digabungkan dengan "Virtual Reality" (reality maya) untuk memberikan ruang pembelajaran yang lebih luas di dalam bidang teknologi.

* Required

GAMBAR 1

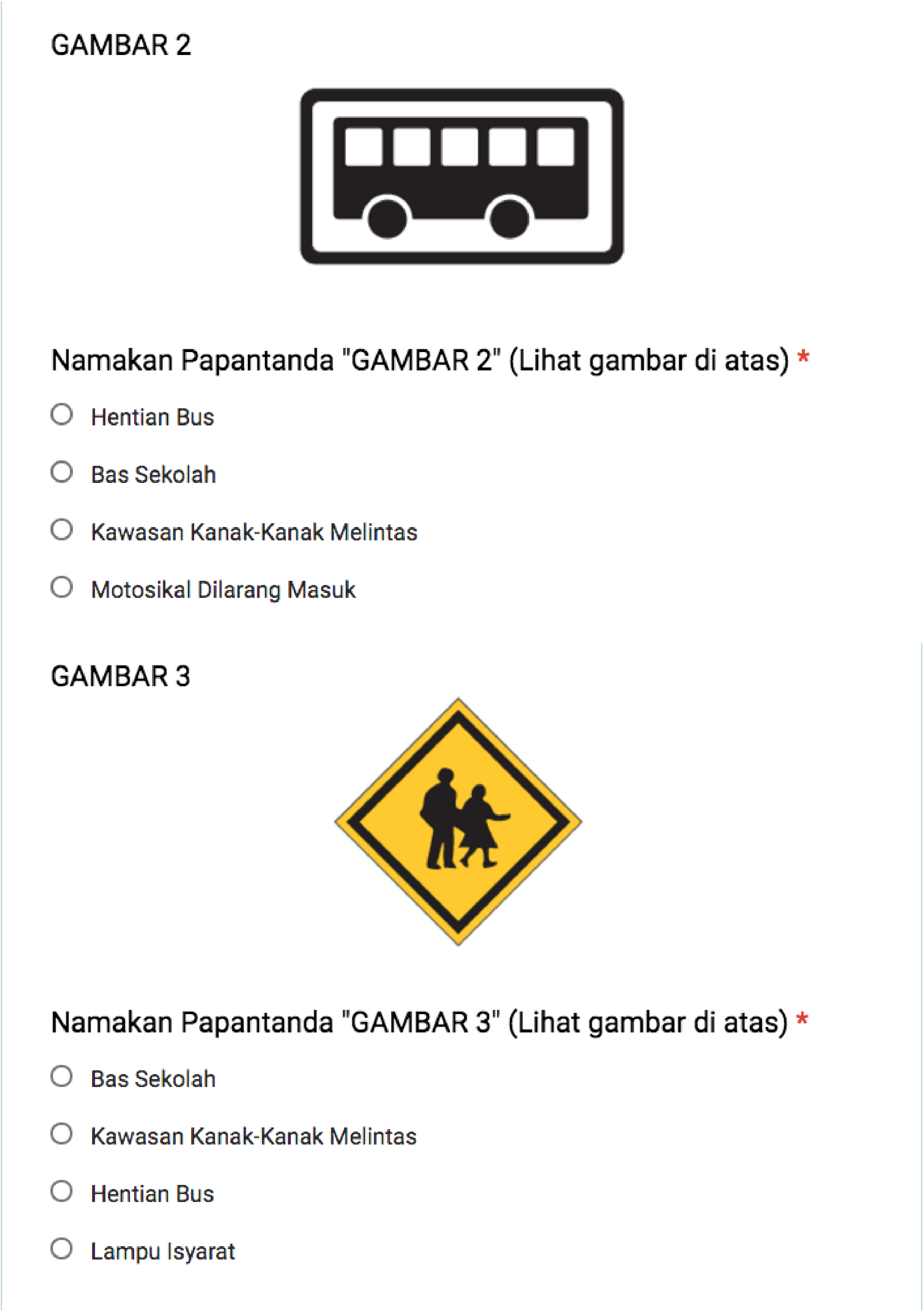
Namakan Papantanda "GAMBAR 1" (Lihat gambar di atas) *

○ Motosikal Dilarang Masuk

○ Hentian Bus

○ Padang Bermain

○ Berhenti



GAMBAR 2

Namakan Papantanda "GAMBAR 2" (Lihat gambar di atas) *

○ Hentian Bus

○ Bas Sekolah

○ Kawasan Kanak-Kanak Melintas

○ Motosikal Dilarang Masuk

GAMBAR 3

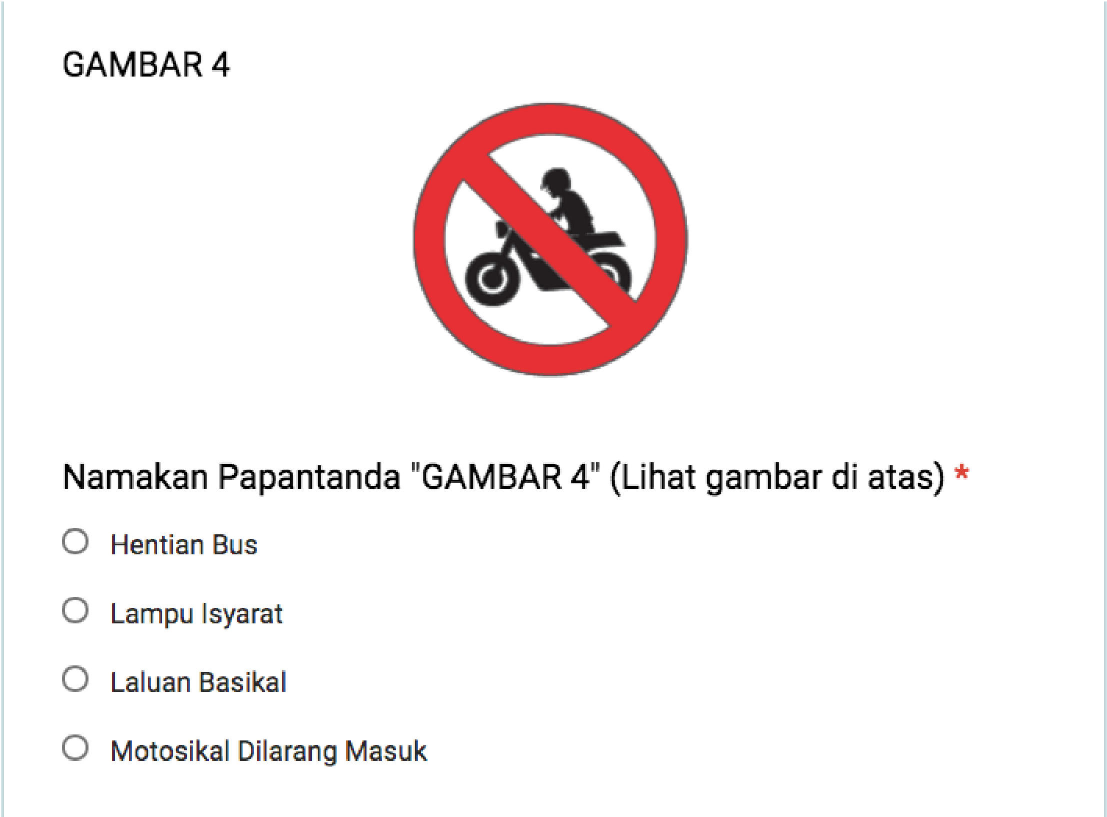
Namakan Papantanda "GAMBAR 3" (Lihat gambar di atas) *

○ Bas Sekolah

○ Kawasan Kanak-Kanak Melintas

○ Hentian Bus

○ Lampu Isyarat



GAMBAR 4

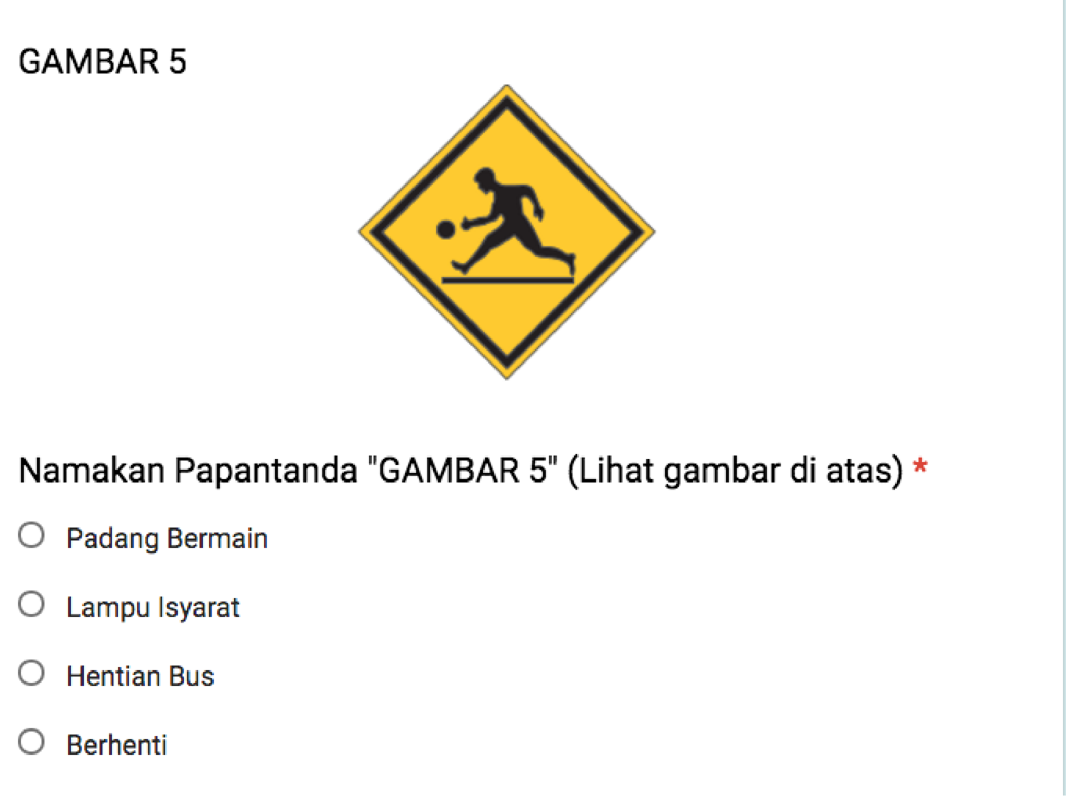
Namakan Papantanda "GAMBAR 4" (Lihat gambar di atas) *

○ Hentian Bus

○ Lampu Isyarat

○ Laluan Basikal

○ Motosikal Dilarang Masuk



GAMBAR 5

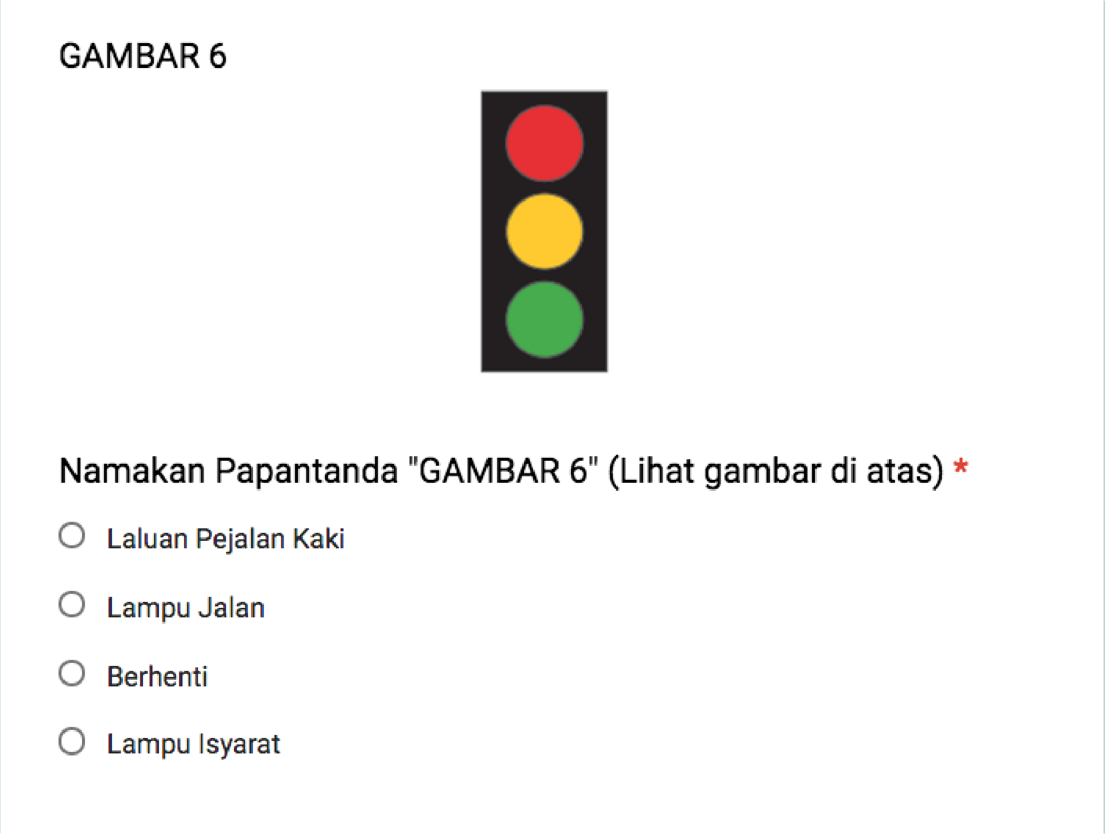
Namakan Papantanda "GAMBAR 5" (Lihat gambar di atas) *

○ Padang Bermain

○ Lampu Isyarat

○ Hentian Bus

○ Berhenti



GAMBAR 6

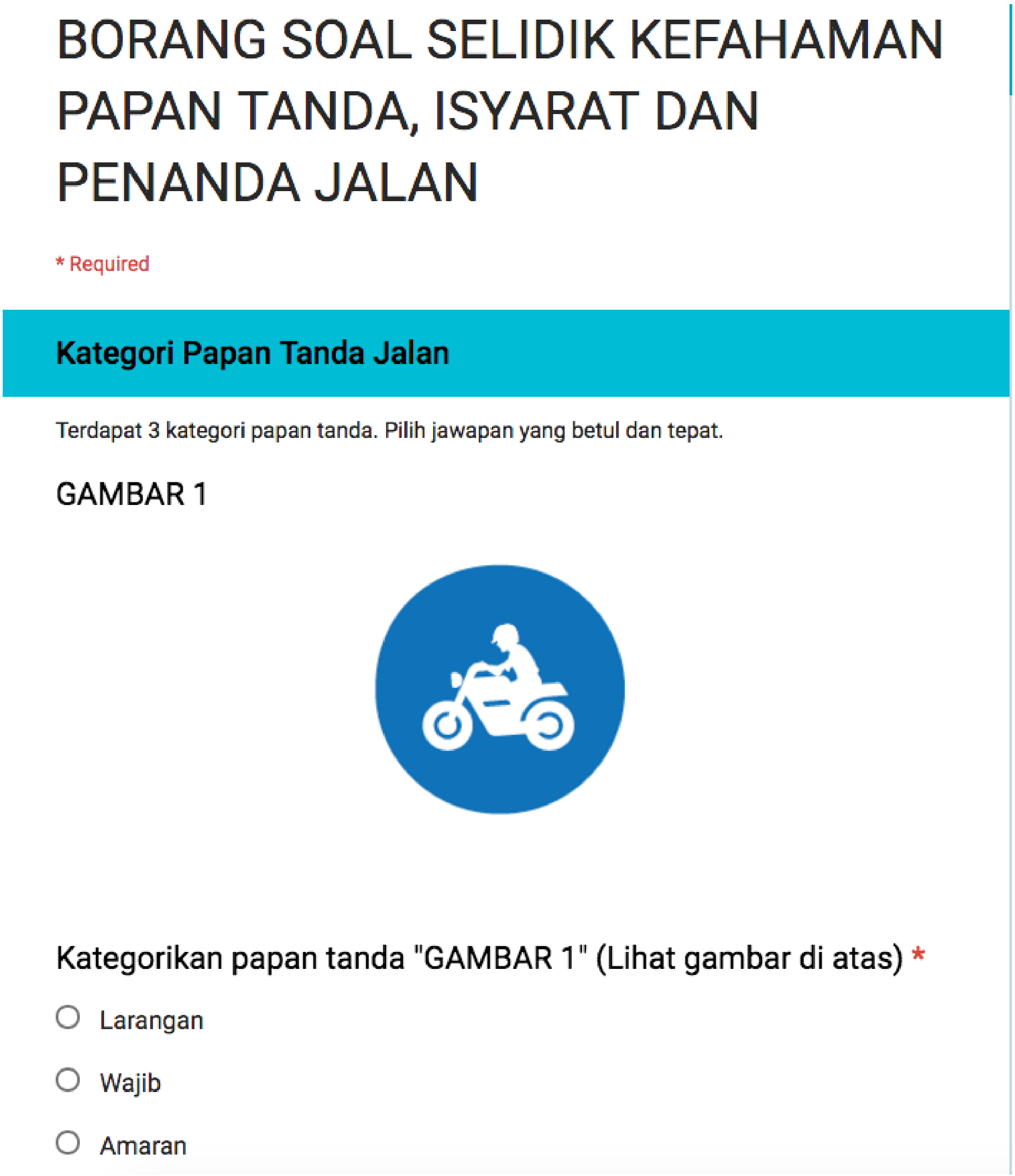
Namakan Papantanda "GAMBAR 6" (Lihat gambar di atas) *

○ Laluan Pejalan Kaki

○ Lampu Jalan

○ Berhenti

○ Lampu Isyarat



BORANG SOAL SELIDIK KEFAHAMAN PAPAN TANDA, ISYARAT DAN PENANDA JALAN

* Required

Kategori Papan Tanda Jalan

Terdapat 3 kategori papan tanda. Pilih jawapan yang betul dan tepat.

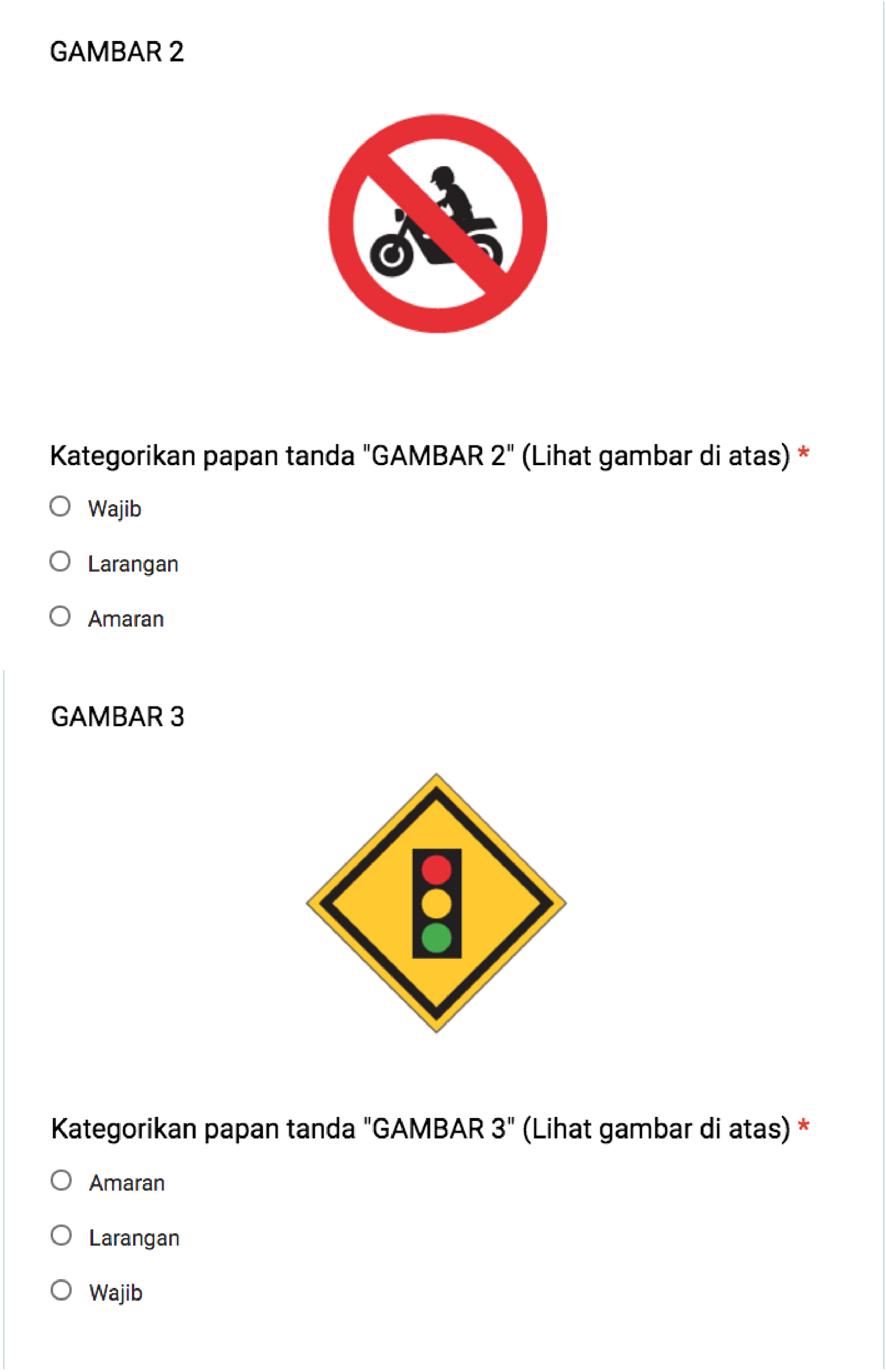
GAMBAR 1

Kategorikan papan tanda "GAMBAR 1" (Lihat gambar di atas) *

○ Larangan

○ Wajib

○ Amaran



GAMBAR 2

Kategorikan papan tanda "GAMBAR 2" (Lihat gambar di atas) *

○ Wajib

○ Larangan

○ Amaran

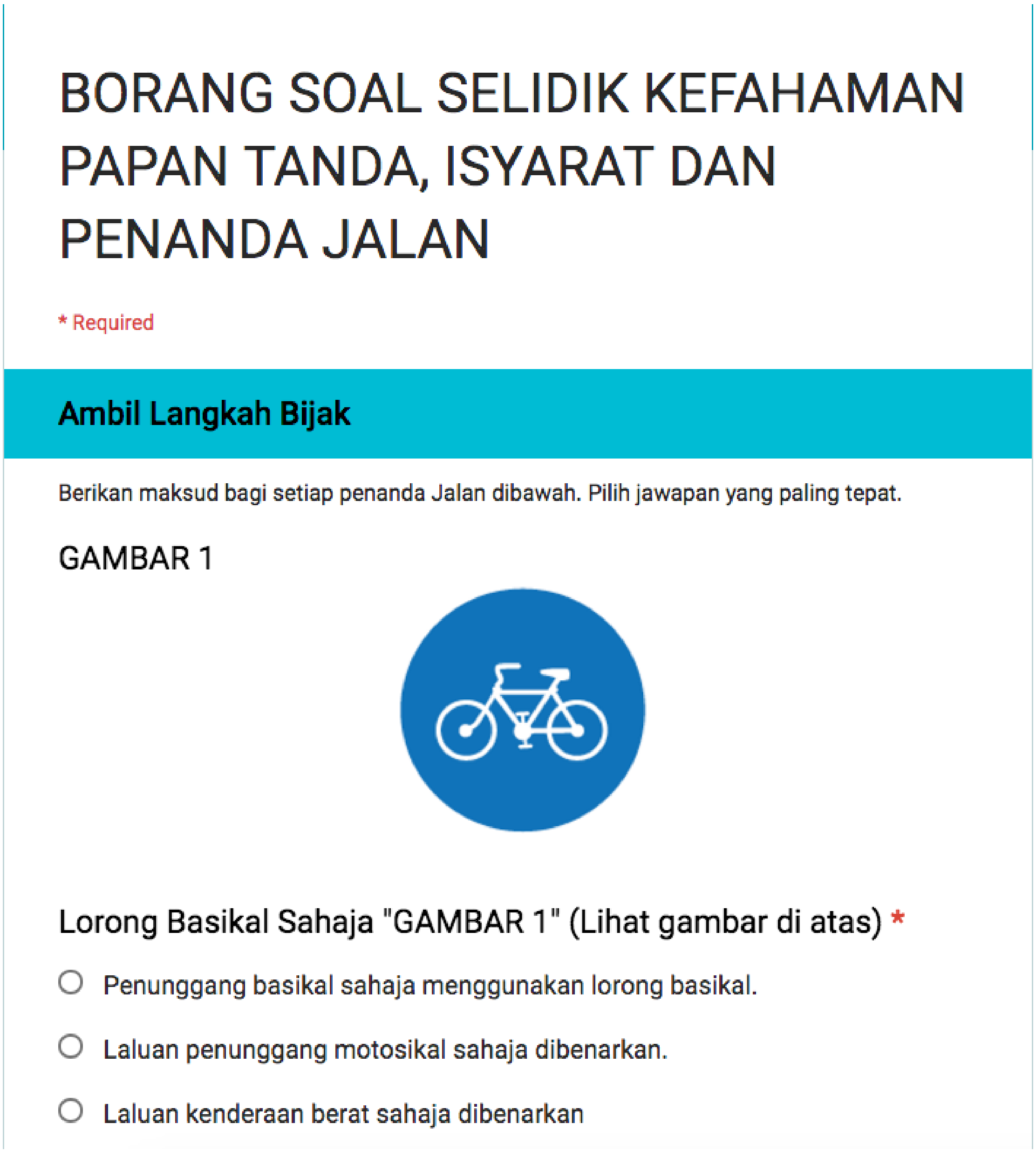
GAMBAR 3

Kategorikan papan tanda "GAMBAR 3" (Lihat gambar di atas) *

○ Amaran

○ Larangan

○ Wajib



BORANG SOAL SELIDIK KEFAHAMAN PAPAN TANDA, ISYARAT DAN PENANDA JALAN

* Required

Ambil Langkah Bijak

Berikan maksud bagi setiap penanda Jalan dibawah. Pilih jawapan yang paling tepat.

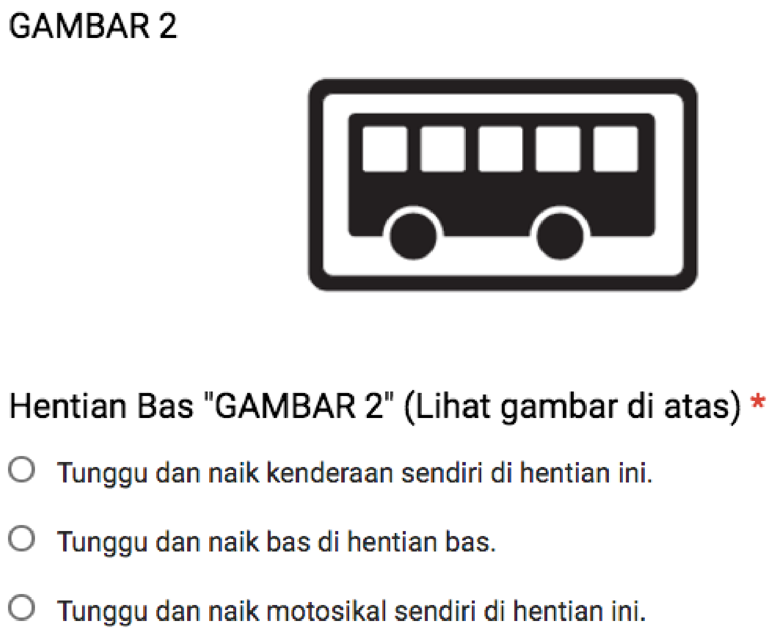
GAMBAR 1

Lorong Basikal Sahaja "GAMBAR 1" (Lihat gambar di atas) *

○ Penunggang basikal sahaja menggunakan lorong basikal.

○ Laluan penunggang motosikal sahaja dibenarkan.

○ Laluan kenderaan berat sahaja dibenarkan



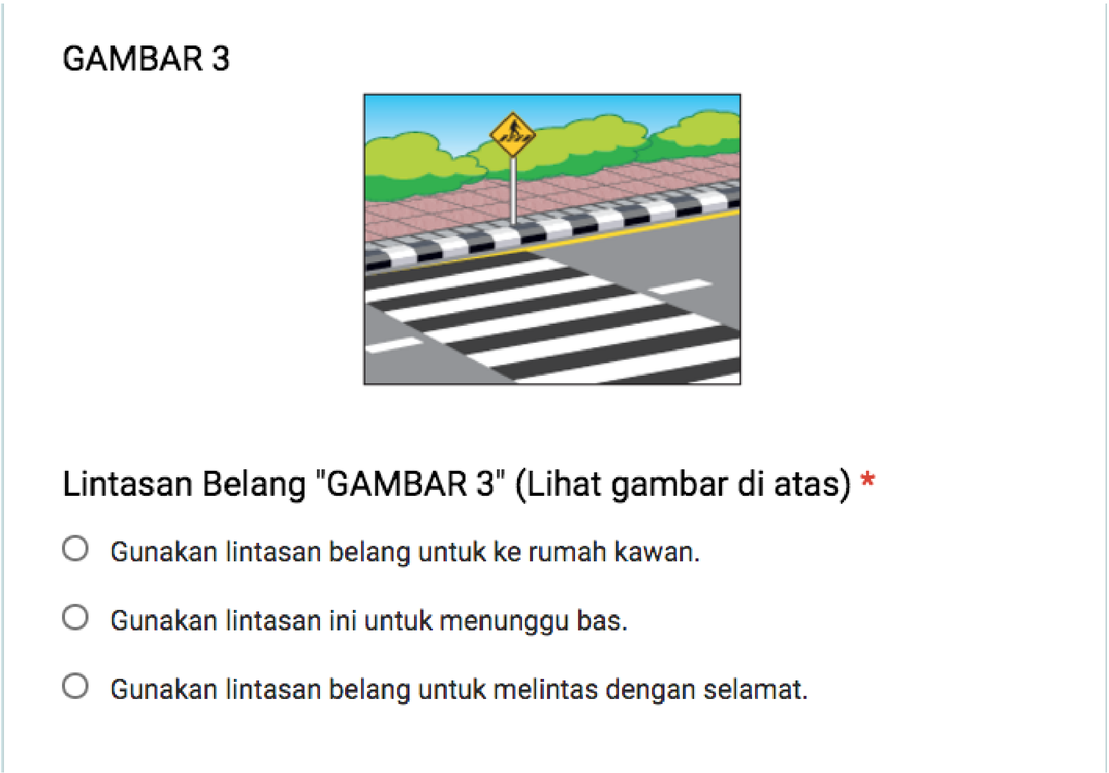
GAMBAR 2

Hentian Bas "GAMBAR 2" (Lihat gambar di atas) *

○ Tunggu dan naik kenderaan sendiri di hentian ini.

○ Tunggu dan naik bas di hentian bas.

○ Tunggu dan naik motosikal sendiri di hentian ini.



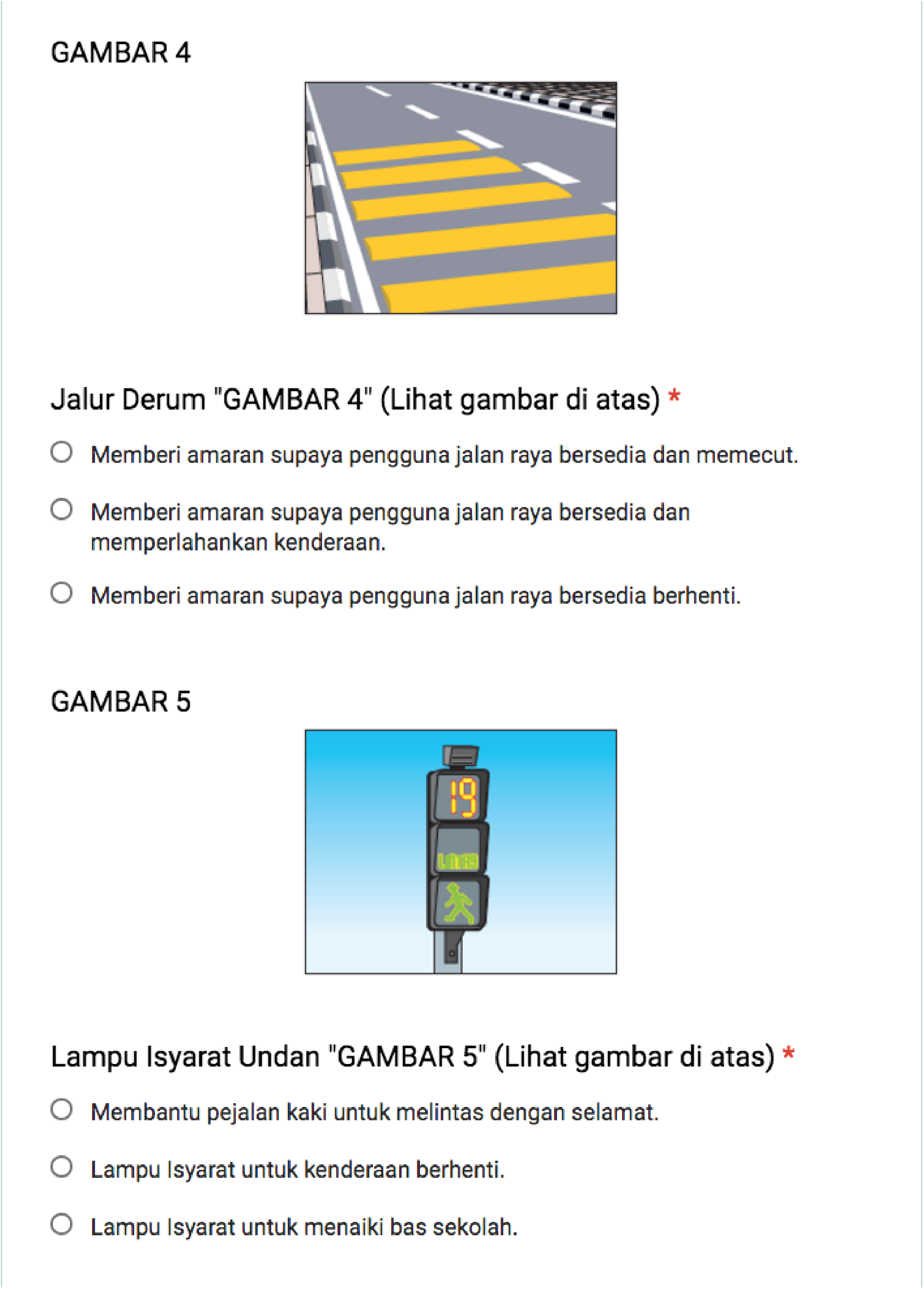
GAMBAR 3

Lintasan Belang "GAMBAR 3" (Lihat gambar di atas) *

○ Gunakan lintasan belang untuk ke rumah kawan.

○ Gunakan lintasan ini untuk menunggu bas.

○ Gunakan lintasan belang untuk melintas dengan selamat.



GAMBAR 4

Jalur Derum "GAMBAR 4" (Lihat gambar di atas) *

○ Memberi amaran supaya pengguna jalan raya bersedia dan memecut.

○ Memberi amaran supaya pengguna jalan raya bersedia dan memperlahankan kenderaan.

○ Memberi amaran supaya pengguna jalan raya bersedia berhenti.

GAMBAR 5

Lampu Isyarat Undan "GAMBAR 5" (Lihat gambar di atas) *

○ Membantu pejalan kaki untuk melintas dengan selamat.

○ Lampu Isyarat untuk kenderaan berhenti.

○ Lampu Isyarat untuk menaiki bas sekolah.