ENHANCED 3D TERRAIN VISUALIZATION PROCESS USING GAME ENGINE

MOHD HAFIZ BIN MAHAYUDIN

MASTER OF SCIENCE (MULTIMEDIA STUDIES)
UNIVERSITI UTARA MALAYSIA
2018

Awang Had Salleh
Graduate School
of Arts And Sciences

Universiti Utara Malaysia

PERAKUAN KERJA TESIS / DISERTASI
(Certification of thesis / dissertation)

Kami, yang bertandatangan, memperakukan bahawa
(We, the undersigned, certify that)

MOHD HAFIZ MAHAYUDIN

calon untuk Ijazah **MASTER**
(candidate for the degree of)

telah mengemukakan tesis / disertasi yang bertajuk:
(has presented his/her thesis / dissertation of the following title):

"ENHANCED 3D TERRAIN VISUALIZATION PROCESS USING GAME ENGINE"

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.
(as it appears on the title page and front cover of the thesis / dissertation).

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : ***01 Jun 2017.***
That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:
June 01, 2017.

Pengerusi Viva: *(Chairman for VIVA)*	**Assoc. Prof. Dr. Fauziah Baharom**	Tandatangan *(Signature)*
Pemeriksa Luar: *(External Examiner)*	**Assoc. Prof. Dr. Mohd Shafry Mohd Rahim**	Tandatangan *(Signature)*
Pemeriksa Dalam: *(Internal Examiner)*	**Assoc. Prof. Dr. Ariffin Abdul Mutalib**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Dr. Ruzinoor Che Mat**	Tandatangan *(Signature)*

Tarikh:
(Date) *June 01, 2017*

Permission to Use

In presenting this thesis in fulfillment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the University Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for the scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

Abstract

Recently, many information visualization regarding terrain use 2D maps which include shading and lines to show the terrain. However, the emerging 3D terrain visualization technologies and software may produce a lot of terrain information. This emerging technology is also concurrent with the growth of game engines. As for this study, Unity3D, one of these game engines, has built-in terrain engine that provides 3D terrain visualization. Moreover, this engine provides the ability to be able to publish as web application for the online environment. Based on the literature review, there are studies related to terrain visualization developed using game engines, however, majority focuses on the capability of terrain visualization in an offline environment. None of these studies focus on the performance of the 3D visualization process in an online environment. Thus, the aim of this study is to enhance the process of generating 3D terrain visualization with GIS data generated from the Unity3D game engine in an online environment. The results of the performance are compared with two different situation that is online and offline. Several experiments are conducted and performances are measured based on loading time, response time, frames per second (FPS), memory usage and CPU usage of different terrain data types and size. The study adopts design research process that is comprised of problem identification from literature review, solution development by using the process to develop the prototype needed, and evaluation by comparing the output of the visualization process. The findings show that the process of enhancing 3D terrain visualization with GIS data generated from the Unity3D game engine in offline environment is better compared to those online. This is due to the compression and the need for Unity3D web player to make contact with the Unity server for authentication and also for visualization during online. Furthermore, operating system resource needs to be used before it goes online. The main finding of this study is the new algorithm of enhancing 3D terrain visualization process using Unity3D game engine. The algorithm can be divided into three processes which are terrain data reading, terrain data conversion, and terrain data processing. It may assist the developer on how to enhance the process of developing web-based 3D terrain visualization using Unity3D game engine.

Keywords: 3D terrain, terrain visualization, game engine, Geographical Information System

Abstrak

Pada masa ini, kebanyakan maklumat tentang bentuk muka bumi menggunakan peta 2D yang menggunakan kaedah teduhan dan garis untuk menunjukkan maklumat bentuk muka bumi. Walau bagaimanapun, kemunculan teknologi visualisasi bentuk muka bumi 3D dan perisiannya boleh menghasilkan banyak maklumat tentang bentuk muka bumi. Kemunculan teknologi ini juga bersamaan dengan perkembangan enjin permainan. Untuk kajian ini, Unity3D, salah satu daripada enjin permainan, mempunyai enjin bentuk muka bumi terbina didalamnya yang boleh menghasilkan visualisasi bentuk muka bumi 3D. Selain itu, enjin ini memberikan keupayaan untuk membolehkan ia dihasilkan sebagai aplikasi web untuk persekitaran dalam talian. Berdasarkan kajian literatur, terdapat banyak kajian yang melibatkan penggunaan enjin permainan bagi menghasilkan bentuk muka bumi, walaubagaimanapun, kebanyakan kajian ini melibatkan keupayaan visualisasi bentuk muka bumi dalam persekitaran luar talian dan tiada kajian yang melibatkan proses visualisasi bentuk muka bumi 3D dalam persekitaran atas talian. Oleh itu, tujuan kajian ini adalah untuk manambahbaik proses penjanaan visualisasi bentuk muka bumi 3D dengan data GIS yang dihasilkan dari enjin permainan Unity3D dalam persekitaran di atas talian. Keputusan hasil daripada prestasi dibuat dengan membandingkan dua situasi berbeza iaitu atas talian dan juga di luar talian. Beberapa eksperimen yang telah dilakukan dan prestasinya diukur berdasarkan masa muatan, masa capaian, bingkai sesaat (FPS), pengunaan memori dan pengunaan CPU pada saiz data yang berbeza. Kajian ini mengunakan proses rekabentuk kajian yang terdiri dari pengenalpastian masalah dari kajian literatur, penyelesaian masalah dengan mengunakan proses bagi membangunkan prototaip dan penilaian dengan membandingkan hasil keluaran dari proses visualisasi. Keputusan menunjukkan bahawa hasil daripada proses penambahbaikan visualisasi bentuk muka bumi 3D dengan data GIS dari enjin permainan Unity3D di luar talian adalah lebih baik jika dibandingkan dengan di atas talian. Ini adalah kerana proses mampatan dan perlunya pemain pelayan Unity3D untuk menghubungi pelayan *Unity* untuk pengesahan dan juga untuk visualisasi semasa di atas talian. Sementara itu, penggunaan sumber sistem pengoperasian diperlukan sebelum ia boleh berada di atas talian. Penemuan utama kajian ini adalah algoritma baru untuk menambahbaik proses visualizasi bentuk muka bumi 3D menggunakan enjin permainan Unity3D. Algoritma ini boleh dibahagikan kepada tiga proses iaitu pembacaan data bentuk muka bumi, penukaran data bentuk muka bumi dan pemprosesan data bentuk muka bumi. Penemuan ini boleh membantu pembangun aplikasi dalam mengenalpasti bagaimana cara untuk menambahbaik proses pembangunan visualisasi bentuk muka bumi 3D berasaskan web menggunakan enjin permainan Unity3D.

Kata kunci: bentuk muka bumi 3D, visualisasi bentuk muka bumi, enjin permainan, Sistem Maklumat Geografi

Acknowledgement

I would like to express my appreciation and gratitude to everyone who has contributed in completing this thesis. It was a pleasure to study under Dr Ruzinoor bin Che Mat supervision. It is not enough to thank him very much for his guidance to achieve my goal. Without his valuable support, my thesis would not have been possible. Also, I would like to express my thanks, Dr Ruzinoor bin Che Mat supervision for his comments which help to improve my work. I would like also to thank my parents and all of my relatives for their love and support. My goal would not have been achieved without them. I dedicate this work to my parents. I am very grateful to everyone that helped in my studies. I also like to thank you Assoc Prof. Dr Mohd Shafry Mohd Rahim, Mohd Naim Shah Bin Abdul Rahim, Mohd Khalid Mokhtar and all from UTM MagicX for their warm welcome given during my visit there as well giving me insight on game engine capability. As well to Jurupro staffs who help in explaining to me on GIS data capture and process to produce the data needed for the research, also to the manager of RISDA Tg Genting, Mr.Mansor b. Awang for his cooperation in completing this study. Also to all examiners, they were very kind during the viva and during the period of the correction. Additionally, their comments have helped to improve this work. I had a very enjoyable study at Universiti Utara Malaysia (UUM). Not only, does it have a beautiful natural environment but the university also has helpful staff. Finally, I would like to thank all of my friends for their encouragement during my study.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

3D:	Three Dimensional
GIS:	Geographic Information System
VR:	Virtual Reality
DEM:	Digital Elevation Model
TIN:	Triangular Irregular Network
DSM:	Digital Surface Model
DTM:	Digital Terrain Model
RAM:	Random access memory
GPU:	Graphical Processing Unit
CPU:	Central Processing Unit
GHz:	Gigahertz
VE:	Virtual Environment
HMD:	Head Mounted Display
UAV:	Unmanned aerial vehicle
MaCGDI:	Malaysian Geospatial Data Infrastructure
VRML:	Virtual Reality Markup Language
CAVE:	Cave Automated Virtual Environment
UDK:	Unreal Development Kit
VE:	Virtual Environment
UDEQ:	Utah Department of Environmental Quality
GCS:	Geographic coordinate systems

PCS: Projected coordinate systems

ESRI: Environmental Systems Research Institute

MS: Millisecond

FPS: Frame Per Second

GMV: Geospatial Modeling & Visualization

CENACARTA: Mozambique National Cartography and Remote Sensing Centre

HIPO: hierarchy plus input-process-output

CHAPTER ONE
INTRODUCTION

1.1 Background

Terrain visualization techniques have been around for years and it can be categorized into the digital and non-digital format. In the earlier years, non-digital terrain visualization was referred to as “map”. This was used to show the location and elevation of the terrains. Although it is effective, it requires certain skills to understand the map information. As technology progress, digital terrain visualization was introduced. The earliest research done on terrain visualization is in the early 90’s where at the time, computers had a decent capability to visualize terrain in three dimensions (3D). In the recent years, as technology advances and computer hardware capacities grew at an exponential rate, the capacity to generate high-resolution 3D terrain visualization has increased. As can be seen in 3DEM (2014) and Cesium (2014).

Terrain visualization uses Geographical Information System (GIS) to digitally display geographical information in computers. Terrain visualization effectively interprets spatial data of earth terrain, showing the earth information digitally. The data it contained is mostly layered to hold different types of information (National Geographic Society, 2014). GIS data such as Digital Elevation Model (DEM), Triangular Irregular Networks (TIN), Digital Terrain Model (DTM) and Digital Surface Model (DSM) is converted 3D model which will have contour and elevation information. Research conducted by Wyld (2010) stated that GIS can be used to promote tourism. Another research conducted by Awadallah, Gehman, Kuttler, and

Newkirk (2004) examined 3D radar information of propagation data for handling ships.

As computer hardware and software advances, a different process of terrain visualization was explored. Rather than creating software from the beginning, there are other methods, such as using a game engine to generate the terrain visualization in some applications. Game engine refers to a set of tools that are combined to simplify the creation of games. A lot of research was conducted on the game engine to see the different ways to utilize it for some real applications rather than just for computer games. The research conducted by Navarro, Pradilla, and Rios (2012) discussed the different types of game engines and highlighted the functionalities of each game engine. Some of the game engines highlighted include Unreal development kit (Epic Games, 2014), CryEngine (CryTEK, 2014), Unity3D (2014), and Torque (GarageGames.com, 2014) could be utilized to generate terrain and other applications. Other than that, terrain visualization also could be generated using VRML, HTML 5 and other software such as Biospehere3D, Cesium, Earth3D, GenesisIV, Hftool, Landserf, and MicroDEM. However, the majority of research conducted in this domain is on the capability of terrain visualization in an offline environment and none of the research examined the performance of the 3D visualization process in an online environment. This study aims to examine the performance of the enhanced process for terrain visualization, which uses Unity3D for terrain visualization in a web environment. That is why the aim of this study is to enhance the process of generating 3D terrain visualization with GIS data generated from the Unity3D game engine in an online environment.

1.2 Problem background

Terrain visualization process is a technique of visualizing terrain data in 3D object thus also brings issues such as the data size needed to visualize it. A study by Hayat, Puech, and Gesquiere (2008) discussed that the process of visualizing terrain data requires numerous 3D vertexes and triangular to generate a 3D terrain. The author also mentions that the requirements needed to visualize such a terrain include DEM data, orthographic data, projection data, as well as the hardware and setup needed to generate the terrain data. Another study by Cowgill et al (2012) explored the terrain data from Haiti earthquake which used a massive amount of hard disk space (67Gb) consist of DEM data acquired from ground-based, airborne data, as well as the hardware used to generate the terrains with the projection used. In another study related to terrain visualization process by Yusuf, Mostafa, and Elarif (2014) proposed that using the processing power of GPU can achieve faster frame rates when visualizing the terrain data. Based on all of these studies on terrain visualization process, it can be concluded that terrain visualization process requires massive computer power in terms of data storage and processing power to generate the terrains. Most of the studies do not mention the performance of the terrain visualization process for generating the terrain in online and offline environments.

1.3 Problem Statement

Terrain visualization is a technique to visualize GIS terrain data into three-dimensional (3D) model that allow users to view terrain in 360 degrees view with X, Y and Z axis. The Malaysian Geospatial Data Infrastructure (MaCGDI) provides 2D maps as means to deliver information to the users however if the visualization of the data is in 3D forms, better information can be delivered to the user.

In recent studies on terrain visualization show that terrain visualization is able to work on multiple platforms, Ruzinoor et. al. (2009) in their research used VRML (Virtual Reality Markup Language) to generate online 3D terrain visualization. Beside VRML there are alternatives to generate terrain visualization by using game engine. Popular game engine such as Torque, Unreal Development Kit (UDK), Unity3D and CryEngine have built-in terrain engine that can assist in terrain visualization, reviewing previous literature reveals that most study on performance of Random Access Memory (RAM), Graphics Processing Unit (GPU) and Central Processing Unit (CPU) on game terrain visualization is conducted in offline environments.

Studies by Yang, Wuensche, and Lobb (2004) and Wyeld (2007) investigated the use of torque game performance in visualizing terrain in offline environment however the study did mention the performance while in an online environment and what the process conducted for terrain visualization.

In another study, Prasithsangaree (2003) and Rathnam, Pfingsthorn, and Birk (2009) studied the performance of terrain visualization in Unreal Engine, however, they did not mention how the terrain visualization process was conducted and the performance of the terrain visualization in online environments.

Study by Dar-Hsiung et al.(2012), Wang et al.(2010), Kang, Kim, & Han (2015) and Beirami, Cho, & Yu (2015) did not mention the performance of terrain visualization as well the process that was used for visualization of the terrain in offline or online environments. Study conducted by Indraprastha and Shinozaki (2009), Jarvis, Løvset, & Patel (2015) and Humbert, Chevrier, and Bur (2011) mentioned about the performance of the terrain visualization but it is in offline environments.

While reviewing possible terrain visualization process from a literature review on Unity3D, Unreal Engine, and Torque, it was discovered that the game engines use the same process in terrain visualization.

This study proposed an enhanced process of terrain visualization inside Unity3D that is different from process currently used in Torque and Unreal Engine and Unity3D. The enhanced process would be tested in online environments as well offline to examine the performance of the enhanced processed by measuring the criteria that were used in previous studies that are FPS, Memory usage, and CPU usage. Two new criteria will be added as in an online environment that is loading time and response time.

The reason to test the enhanced process is to examine how the enhanced process helps in terrain visualizations inside Unity3D. GIS data usually is big and by using the enhanced process and test the performance of the enhanced process we can understand what the performance of the enhanced process in the online as well offline environment.

Unity3D is a free game engine and easy to use with support to it asset stores premade projects and programs examples are made available for the user to explore as well huge resource of documentation given thus reducing the time for the new user to learn about Unity3D.

1.4 Motivation

Terrain visualization allows more depth in understanding of the surface, allowing more information to be demonstrated. This statement is supported by Tateyama, Oonuki, Sato, and Ogi (2008). The above-mentioned authors expressed that terrain visualization could also be applied in Cave Automatic Virtual Environment (CAVE) system environment, whereby it can display information about seismic data of the Pacific Ocean and Philippine Sea plate data. With this capability, it can help the user understand the relationships between all of these data.

Terrain visualization also helps in planning since more information can be demonstrated. A study by Hagedorn and Döllner (2007) used terrain visualization with a 3D building to help in emergency situations like a fire scenario. Terrain visualization can also help in city planning and has the capability of working in an online environment.

The game engine also could be utilized in generating 3D terrain visualization. A study by Friese, Herrlich, and Wolter (2008) revealed that by using three different types of the game engines which include Quake3, CryEngine and Unreal Tournament can visualize a CAVE environment together with terrain information. Game engine allows the researcher to achieve a faster result as much basic functionality. It also allows easier exploration of the terrain information and more understanding of the terrain data.

1.5 Research Questions

The problem statement discussed leads to the main research questions of this study, which is:

i. What is the terrain visualization process involved in generating the real-world terrain inside the game engine?

ii. What is the performance of enhanced terrain visualization process using a game engine in offline and online environments?

1.6 Research Objectives

The main objectives of this research are to verify terrain data performance inside the game engine specifically Unity3D. Specific objectives of this research are:

i. To enhance the process of visualizing 3D terrain using the game engine.

ii. To develop a prototype with the proposed enhanced 3D terrain visualization process in the game engine.

iii.. To evaluate the performance of the proposed enhanced 3D terrain visualization process in online and offline environments.

1.7 Scope of the Study

This studies scope is to test single game engine i.e.Unity3D. Testing is to be done using oil palm plantation terrain data which consist of four different sizes i.e. 16.927292 hectares, 5.49895 Hectares, 2.34673 Hectares and 0.841018 Hectares.

The experiments would be done to test the performance of the prototype that uses enhanced the process to generate 3D terrain visualization in the online and offline environment. The formats of the data are float (FLT) and header (HDR).

1.8 Significance of the Study

The main purpose of this study is to use the processes inside the game engine as a terrain visualization tool to provide an alternative to the current free and commercialized 3D terrain visualization software. Game engine provides alternative functions that can be included as plug-ins, thus necessitating the researcher to study the numerous variety of data. Additionally using game engine allows more user interaction with the terrain and enable experiencing the terrain from different perspectives. This also highlights the importance of terrain visualization in Malaysia especially since terrain visualization studies in Malaysia are scarce, this would help promote 3D terrain visualization in Malaysia. Furthermore, the study also intends to help provide alternatives for researchers or explorers to view a terrain in the 3D model especially for locations that are dangerous or hard to reach. In addition to that, enable the creation of simulations with no trouble using game engines, also contributing to the bodies of knowledge. This research focuses on the performance process of terrain visualization using the game engine which is Unity3D.

1.9 Theses structure

The structure of this thesis is organized into six chapters.

Chapter one explains the research background on what terrain visualization is and the current trends in terrain visualization. This chapter also provides the required research questions, problem statements, as well as the research objectives that are to be achieved by the end of the research.

Chapter two studies related literature related to terrain visualization, a general overview of visualizations and how terrain visualization relates to conveying information better. This chapter also explains what GIS is, its background, as well as how terrain visualization begins from GIS. This chapter also explored several related studies on GIS usage, and finally described how game engines have been used for terrain visualization in recent years. It also highlights how the combination of both Game Engine and terrain visualization help in conveying information better.

Chapter three explained the methodological aspect applied in this study. The chapter also detailed out the requirements needed for testing and evaluation phase of this study. The method that will be utilized as well as the comparisons to be implemented is also detailed out in this chapter.

Chapter four presents the development aspect of the prototype, how the tests were conducted, as well as provided detailed information on how the data collections process was implemented. The chapter also presents the terrain data processing to be viewed in the Unity3D game engine. It also detailed out the process used to visualize the terrain data inside Unity3D and show the flow of the process.

Chapter five presents the results and discussions after the requirement for data collection needed for the development of the prototype have been fulfilled in chapter four. The result will be divided into seven parts which include loading time, response time, frame per second, CPU usage, memory usage, data size with different situation and justifications from previous studies.

Chapter six summarizes the results and findings that were gathered from chapter five. As well as proffering conclusions on the results, how it contributes to the body of knowledge and explaining how this study would be beneficial for future research.

1.10 Summary

This chapter explained the introduction to this study by explaining the background of the study, the problem background, and the problem statements. This chapter also explained the motivation of this study which drove the research questions and objectives that come with the questions. The chapter also outlined the necessary scope of the study, and the significance it will have. This chapter also details out the theses structure which highlights the content of this theses.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

The online 3D terrain visualization and GIS have an important role in this new era of geo-browser. Nowadays, many activities depend on geo-browsers like Google Map to guide the users in the right direction and also provide the user with the terrain visualization data. This kind of system provides 3D terrain visualization capabilities despite it being limited to areas of high elevation only. Other than geo-browser, this new era also involves the utilization of game engine for generating the 3D terrain visualization with GIS data which most of the applications do not have the online capability. This chapter discusses details on issues regarding visualization, virtual reality, game engine, GIS, terrain visualization process, terrain visualization software and theories related to this study.

2.2 What is Visualization ?

Visualization can be defined as the representation of mental images or data in pictorial or graphical format. It helps for the decision to be made based on analytics presented visually on difficult concepts or new patterns. The concept interactive visualization is enhanced a step further by using technology to drill down into charts and graphs for more details. This interactively changes what data can be view and how it's processed. This is because; the human brain easily grabs and process information from graphs and charts rather than reports or spreadsheets. Charts and graphs assist in visualizing complex data in a simplified way easily (SAS Institute Inc, 2017). The following sub-section discusses the benefits of visualization.

2.2.1 Challenges and Benefits of Visualization

The following section has mentioned several challenges and benefits of information visualization as the gateway to knowledge system and as a pedagogical tool in the humanities (Rutgers, 2017).

1. Information Visualization As The Gateway To Knowledge Systems

Information visualization is an original tactic to produce visual “maps” of abstract information, presenting otherwise vague data in a way that cultivates understanding and recall the information. The familiar forms of digital information visualization are charts and graphs. The rapid advancement in technology transformed visualization to act as a bridge connecting experts and researchers by presenting the augmented information.

2. Information Visualization As A Pedagogical Tool In The Humanities

The interactivity of modern information visualization tools assists users at all levels to engage more deeply with materials in a variety of contexts. There are several advantages of information visualization. Information visualization for the learner combines well with simulation and facilitates collaboration. Simulated visualization allowed dangerous experiments or situations to be presented through simulations. Visualization assists collaborative research by providing a simplified way to convey information that has been gathered previously to a different group of the researchers. In addition to that, visualization uncovers real systems by providing help in unfolding complex information in a simplified way to understand the system, as well as disclosing information that was delicate to observe. Modern visualization tools available help in engaging users to the information's given in various contexts.

2.2.2 The Visualization Process

The visualization process involves the process of changing the data into something that can be easily interpreted from data which can be a collection of numbers or Figure that is yet to be given context. Thus visualization helps in making data much easier to understand. In this context of terrain visualization, data is usually in the form of digital format that is in float and integers thus visualization help in viewing the terrain data. The process of visualizing terrain data requires an algorithm to read the raw terrain data into something that can be visualized in a form of 3D objects. The visualization process also can be view by using the Virtual Reality (VR) technology together with GIS data and could be visualized by utilizing game engine technology.

2.3 Virtual Reality (VR)

VR is a contemporary technology, which is able to mesmerize the entire technological world by its outstanding uniqueness. The following subsection briefly explains VR definition, history, and types of VR. Furthermore, a discussion of VR applications that are applied in varying domains such as architecture, medical simulation, entertainment, and training was also discussed. Virtual Reality is also known as Virtual Environment (VE). Mazuryk and Gervautz (1996) explain that it is a technology which offers some immersive environment experience for its user. Virtual reality is a method of defining virtual world inside a computer by using tool that enables the user to interact with the avatar and environment inside the virtual world. VR is able to replicate a virtual environment as it does in a real-world environment and provides a multi-sensory experience to its user. Although there is a variation of VR applications across several domains, they all sharing similar features like the ability to allow its users to view the three-dimensional images. The main objective of this VR is to go beyond and experience the cyberspace by interacting with its virtual world environment. According to Burdea and Coiffe (2003), VR can be categorized into three I's which are Immersion, Imagination, and Interaction. While Sherman and Craig (2003) in their book titled "Understanding Virtual Reality: interface, application, and design" found four key elements of VR which is, the virtual world, immersion, sensory feedback, and interactivity. VR has been applied in varying domains after its potential was acknowledged and recognized by the researchers. Despite VR not being a new technology to the technology world, its achievements are still in its initial state because; its enormous potentials have attracted countless research and explorations on VR technology. Currently, VR is well known in education, medicine and training domains. VR in education was well supported and encouraged by the experts because it offers students a collaborative learning experience. Consequently, students are now able to interact and learn some complex terms, theories which were difficult to understand through conventional teaching methods. The main aim of including VR technology in education is to offer an effective, attractive and interesting way of instructional delivery of teaching and learning (Klopfer, Osterweil, Groff, & Haas, 2009). This technology can be applied in the training field of driving vehicles, and in complex machinery to avoid any major accident. There are many other advantages of this technology that can enhance safety in the current living standards as compared to contemporary alternatives.

2.4 Game Engine

Game Engine is a tool that helps game creators by reducing the workload of creating games conventionally (Michaelenger, 2013). Current game engines component consist of Inputs, Graphic, Sounds, Networking, Physics, Graphical User Interface (GUI) and Scripts (Michaelenger, 2013). Game engines can also be seen as a framework that helps the developer in tasks such as graphics rendering, sounds and GUI (Alexander, 2014). Game engines have contributed to entertainment, education and medicine. Game engines like Unreal development kit (Epic Games, 2014), Unity3D (2014), CryEngine (CryTEK, 2014) and Torque (GarageGames.com, 2014) have created countless games for entertainment both online and offline. Combining the elements of game engines and GIS data, 3D terrain visualization can be achieved. The game engine enables users to interact with the environments intensely and enable the user to view the 3D terrain through the internet.

2.4.1 Architecture of A Game Engine

Every system has its very own architecture. This is also the same for game engines which helps in creating games much efficiently. Gregory (2009) explains that game engine consists of a few major parts which are programming, human inputs, rendering engine and real-world logic. Lewis and Jacobson (2002) presented the functionality of game engine architecture as shown in Figure 2.1. Game engine architecture consists of a few parts that combine into a single entity, offering numerous functionalities for the creation of games and applications much easier.

The basic structure can be divided into three major parts which are human interaction, design and rendering, and lastly real world logic.

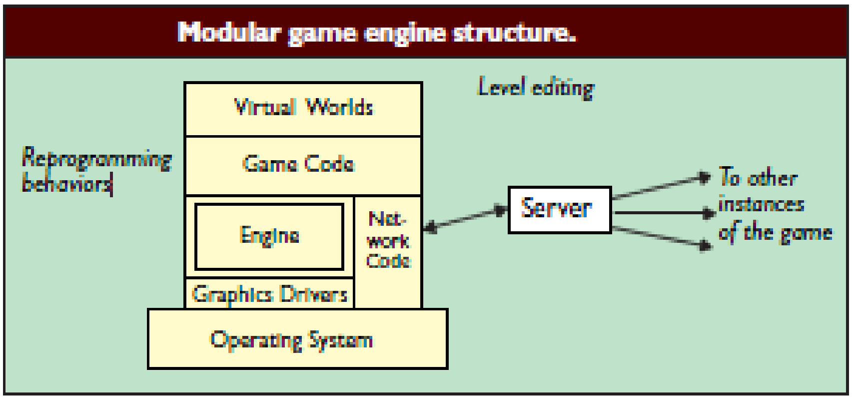

Figure 2.1. Structure of a game engine.

I. Human interaction

Each game would consider how the users interact with the game. Thereby, providing a suitable user interface to allow a better understanding of the how the game would work. Figure 2.2 shows Unity3D GUI.

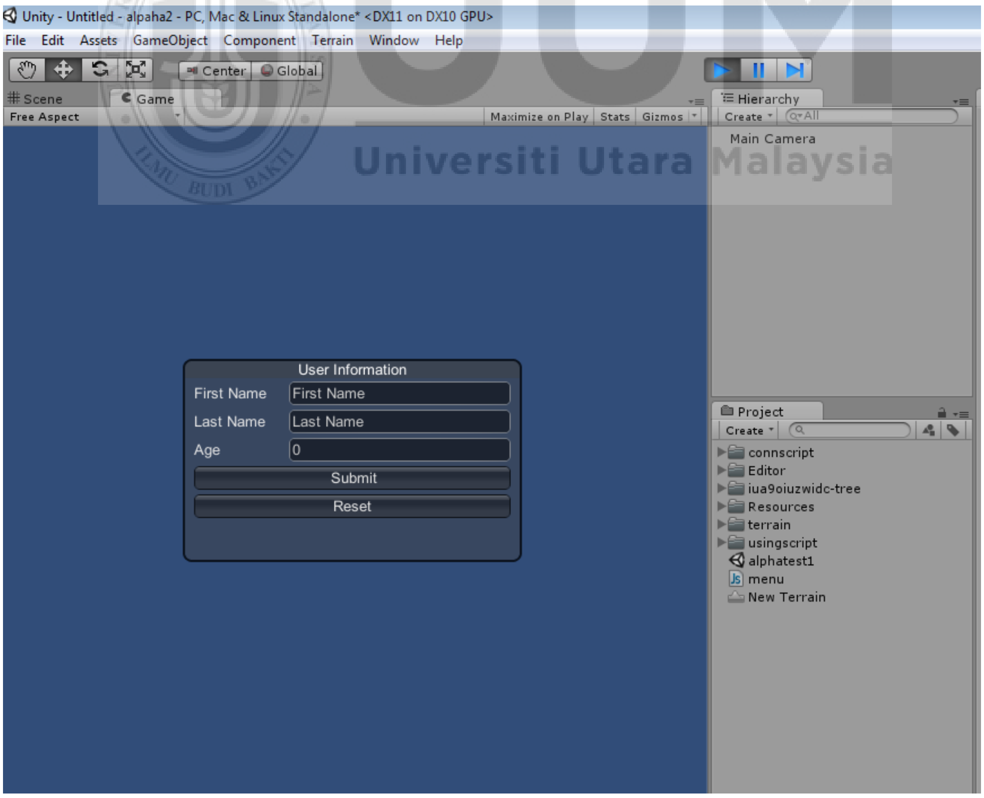

Figure 2.2. Unity3D GUI

II. Design and rendering

This is where the model and environment is created. The model would be given human-like characters such as the face, body, and gestures. In this part, the 3D model would have given biped or bones to support human movement characteristics. Figure 2.3 shows the basic structure of a biped that is not merged into a humanoid 3D model.

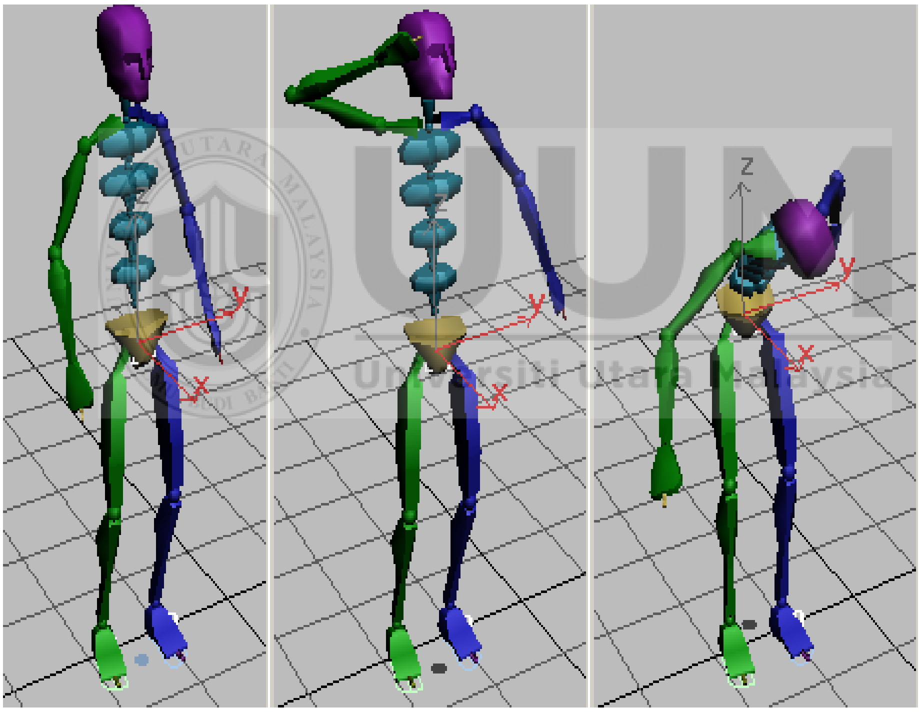

Figure 2.3. Basic Structure of Biped In 3ds Max (Source: Autodesk (2013))

III. Gaming Logic

This part looks at gaming logic such as collisions, physics of the environment. Trigger action when activating a button or a panel. Hodges (2001) explains that gaming logic can be divided into a few categories which are Logical Games, Semantic Games for Classical Logic, and Semantic Games with Imperfect Information.

Logical games refers to two players playing a game which would have an outcome of winning and losing but is determined by how the game is played, the strategy that is used during the game, and how the rule of the is adhered to, would ultimate determine the outcome of the game. Semantic Games for Classical Logic are games that combine a collection of other objects to function properly. This can be seen in the most games where every object in the environment needs to be obtained for it to function. While Semantic Games for Classical Logic requires each object to work properly; Semantic Games with Imperfect Information does not require every object to be obtained for it to function, but will display the information regarding the non-important object despite being fully functional. The department of computer and information science, University of Pennsylvania explains that every logic game is computable (Japaridze, 2014) and its application is very broad as seen from previous research. There are other game engine architectures such as Unity3D.

2.4.1.1 Unity3D

Unity3D is a game engine software created by Unity3D Technologies (2014). A Unity3D game engine is able to develop both online and offline games. Unity3D is capable of providing a wide range of assets available in the unity asset stores to help professionals and new developers to develop games. It has a wide range of documentation and tutorials. Moreover, Unity3D is able to use the concept of the plug-ins, allowing the user to import or just copy the asset file into a specific folder. Unity3D uses JavaScript, C#, and Boo as the main programming language. Unity3D's main advantage is that it could be run on multi-platform such as Window, Macintosh, and Linux and also on mobile platforms. In addition to that, Unity3D is also capable of importing another 3D model like as fbx, sbx, and obj. The Unity3D framework was developed by Wang et al. (2010) (refer Figure 2.4). The framework consists of two main categories: outside Unity3D, and inside Unity3D. Outside Unity3D consist of preparing data such as data of the terrain, 3D models, multimedia contents, satellite UAV images to be sent to game engines. While the work inside Unity3D consists of setting the environment based on the data collected from outside Unity3D, this is where the structure of the model environment is set up by applying content that was collected from outside Unity3D based on needed requirement.

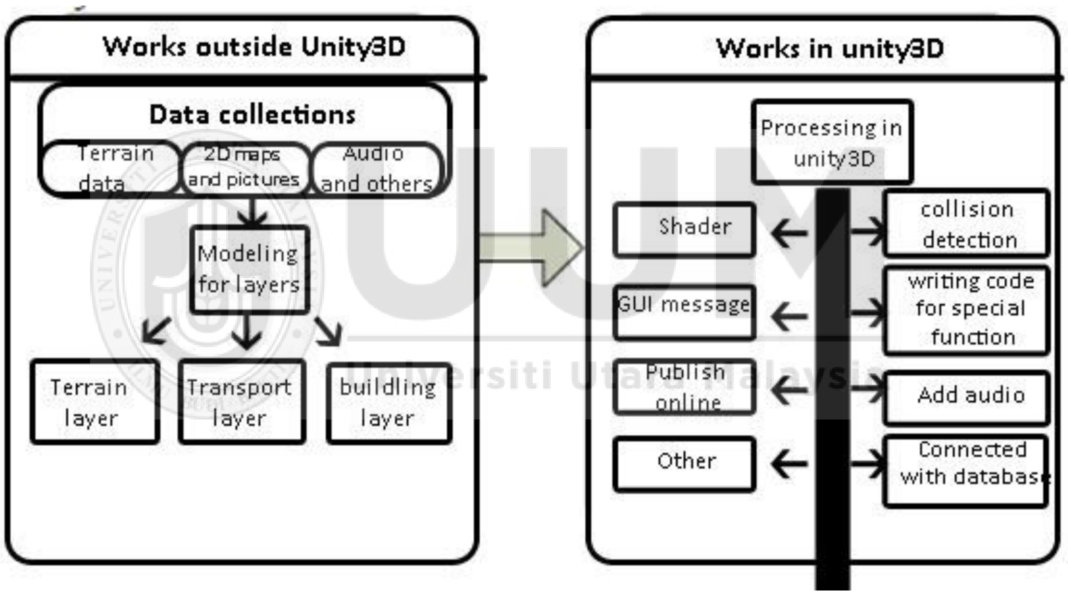

Figure 2.4. Unity3D Game Engine Framework (Wang et. al. (2010))

In a recent study by Messaoudi, Simon, & Ksentini *(2015)* explained that there are a lot of modules implemented in Unity3D architecture. However, the aforementioned authors highlighted six core modules in Unity3D that includes AI, physic, scripting, input, multimedia rendering and networking (refer Figure 2.5). They also mentioned that most recent game engine uses GPU as its core rendering module. The results of this study will be based on Unity3D performance in term of GPU and CPU testing from Unity3D asset store.

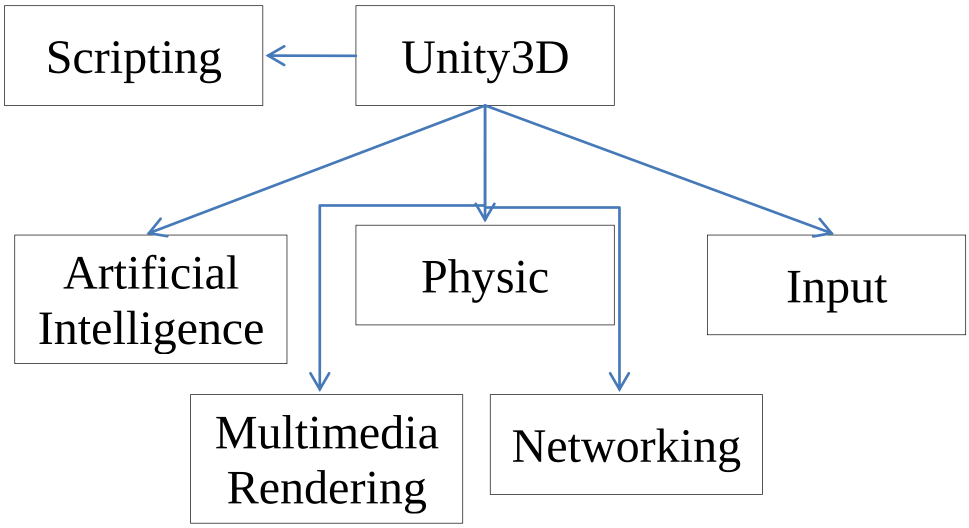

Figure 2.5. Unity3D Six Core Module *(Messaoudi, Simon, & Ksentini, 2015)*

2.4.2 Utilization of Game Engine in Different Sector

As technology advances, new approaches to the virtual environment and its real-world applications are achieved. Game engine provides the means to visualize these situations. Game engine helps the developers to emulate situation much efficiently with the help of tools available inside the game engine. There are several sectors that have benefited from game engines. In a study from the agricultural sector by Maa, Yang, Chen, Zhu, and Guo (2012) utilized Unity3D game engine to emulate the training of the use of agricultural machinery. The authors showcased the differences of each machinery and controls in their simulations on different terrains. Chen, Wang, Zhao, Niu, and Zhu (2010) in their research also utilized game engine in their research to simulate maize farming scenarios with the climate control scenario. Game engines also benefit the security forces as can be seen in a study by Janus research (2014) and Real Visual (2014), focused on military training and on visualizing environment such as oil rig. Game Engine also benefits the construction industry as can be seen in a study by Martin, Chevallier, and Monacelli (2016), which used Unity3D as a tool to help construction workers understand the information regarding buildings. It can also be seen from the work of Humbert and colleagues on the reconstruction of the old city model in the 3D environment using Unity3D (Humbert, Chevrier, & Bur, 2011). Game Engine also contributes in urban planning and architecture as can be seen from Berger and Cristie (2015) in their development of computational fluid dynamics (CFD) tools in Unity3D that identifies the wind and water flow in urban city development. Indraprastha and Shinozaki also utilized Unity3D to design 3D environments of a city (Indraprastha & Shinozaki, 2009).

2.4.3 Game Engines on the Market

There are a lot of different types of game engines available. Each of them has its advantages and disadvantages. Table 1 presents the available game engines used to develop different types of games. It also shows the popularity and availability of the game engines that can be used to develop 2D and 3D games. In addition to that, the table also explains the capabilities and compatibilities of the game engines on varying platforms. Most of the game engines in Table 1 show that PC is a very popular platform for these game engines. These engines are not limited to offline or online, however, plug-ins enable the game engine to be used both online and offline. In Table 2.1, a list of online and offline games created by each game engine is outlined. Most of the game engines listed here own a license for them to be fully functional like exporting into different platforms. Marmalade game engine requires the developer to include marmalade image in their product and Rapid2D allows the game to be created in window 8 environments only. The developer of Unity3D, however, allows a fully functional game engine with a personal license, and it can be published to PC, Web player and recently Android platform. Examples of online games from the Table are Call of Duty: Black Ops – Zombies, World of Tanks and Grandia online. These game engines also provide support for mobile development such as Android and IOS environment gaming.

Table 2.1

The top 14 game engines the market

Title	**Company**	**Platforms**	**Used in**
App Game Kit (P)	The Game Creators	Android, iOS, Mac, PC	Cannon Ball, Hide It Find It, Jumping Jack
BigWorld (P)	Wargaming	Browser, PC	World of Tanks, Grandia Online, Heroes: Scions of Phoenix, Moego, Realm of the Titans
BlitzTech (P)	Blitz Games Studios	3D S, Android, iOS, Linux, Mac, PC, PS3, PS Vita, Wii, Wii U, Xbox 360	House of the Dead: Overkill, Rayman Raving Rabbids, Puss in Boots, KumoLumo, Vitalize
CryEngine 3 (P)	Crytek	Next-gen consoles, PC, PS3, Xbox 360	Crysis, Aion, MechWarrior Online, Sniper: Ghost Warrior 2, Cabal Online 2
GameBryo (P)	Gamebase USA	Android, iOS, PC, PS3, Wii, Wii U, Xbox 360	Catherine, El Shaddai, Epic Mickey, Rocksmith, Warhammer Online: Age of Reckoning
GameMaker (P)	YoYo Games	Android, Browser, iOS, Mac, PC, Windows Phone	Hotline Miami, MrKaroshi, Reflexions, Spelunky
HeroEngine (P)	Idea Fabrik	PC	Star Wars: The Old Republic

Table 2.1 continued

Havok Vision Engine (P)	Havok	Android, iOS, Linux, Mac, PC, PS3, PS Vita, Wii, Wii U, Windows Phone	Arcania: Gothic 4, Carnival Island, Orcs Must Die!, The Settlers 7, Soul Worker
Infernal Engine (C)	Terminal Reality	3D S, Android, iOS, Linux, Mac, PC, PS3, PS Vita, Wii U, Xbox 360	Ghostbusters: The Video Game, Kinect Star Wars, The Walking Dead: Survival Instinct
Marmalade (FS)	Marmalade	Android, BlackBerry OS, iOS, Mac, PC, Smart TVs, Windows Phone	Call of Duty: Black Ops – Zombies, Cut the Rope, Draw Something, Pro Evolution Soccer, Talisman
Rapid2D (FW8)	Rapid2D	PC	Keep Calm and Kill Aliens, London Breaker, Royal Pigeon
Shiva (FWO)	Stonetrip	Android, BlackBerry OS, Browser, Flash, iOS, Linux, Mac, PC, PS3, Wii, Windows Phone, Xbox 360	Babel Rising, NonFlying Soldiers
Unity3D (FR)	Unity3D Technologies	Android, Browser, Flash, iOS, Linux, Mac, PC, PS3, Wii U, Xbox 360	Bad Piggies, Castle Story, Dead Trigger 2, République, Wasteland 2
Unreal Engine 4 (P)	Epic Games	Consoles (TBA) , PC,	Gears of War, Infinity Blade, Mass Effect,

	PS4	Dishonored, BioShock Infinite

*(P) =Pay To Use (C) = Closed down (FS) =free with splash screen (FW8) =(free for window 8 only) (FR) =free with revenue limit and splash screen Written By Aaron (2013)

2.5 GIS

GIS is a system used to capture, store, manipulate, analyze, manage and present all types of geographical data in a form that is easy to understand (ESRI, 2014). GIS has four basic capabilities of handling geospatial data which include: data capture and preparation by referring to data collecting using UAVs, satellite capture imagery, data management including storage and maintenance, and finally data manipulation, analysis and data presentation (Huisman & By, 2009).

2.5.1 History of GIS

GIS started in the 16th century when two renowned French mathematicians: Fermat (2014) and Descartes (2013), found the relationships between graph lines and coordinates system in their philosophy. And in the 17th century, Louis-Alexandre Berthier (2013) used the overlay technique to sequence military strategies. Choropleth map (Oxford University Press, 2014), the map uses the shaded area to represent data and statistical information. Cartogram (Oxford University Press, 2005). United States Department of Agriculture created a choropleth map to show information on farm high-speed internet access by each state shown in Figure 2.6.

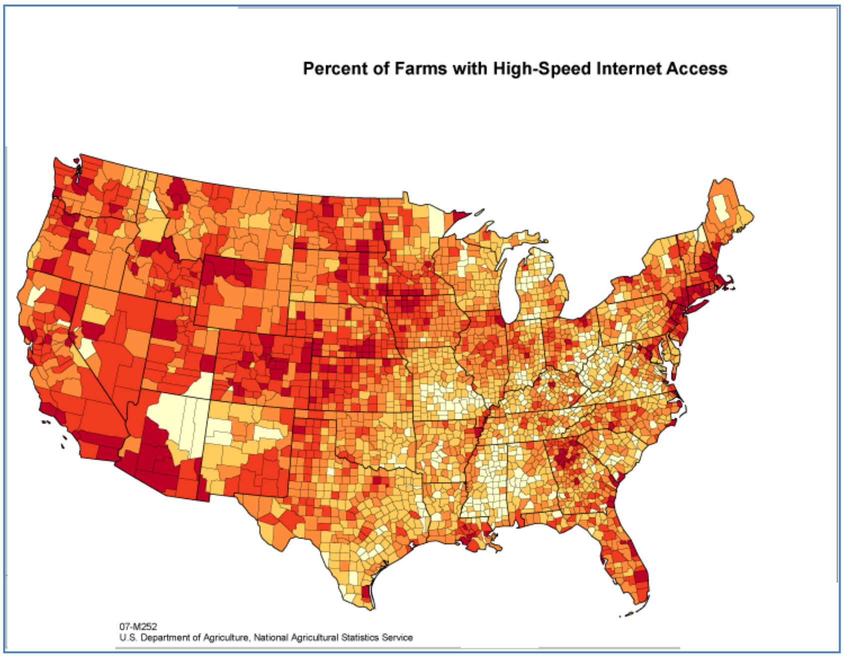

Figure 2.6. Choropleth Map On Internet Access From USDA (2012)

Modern GIS has gone through a lot of research and development since the 1960's. Harvard Lab for Computer Graphics and Spatial lead by Howard Fisher which was dissolved in 1991, created SYMAP among the earliest GIS software. SYMAP utilizes vector image to function as shown in Figure 2.7, the map contains lines and dots with symbols and legends to represent information (ESRI Press, 2005)

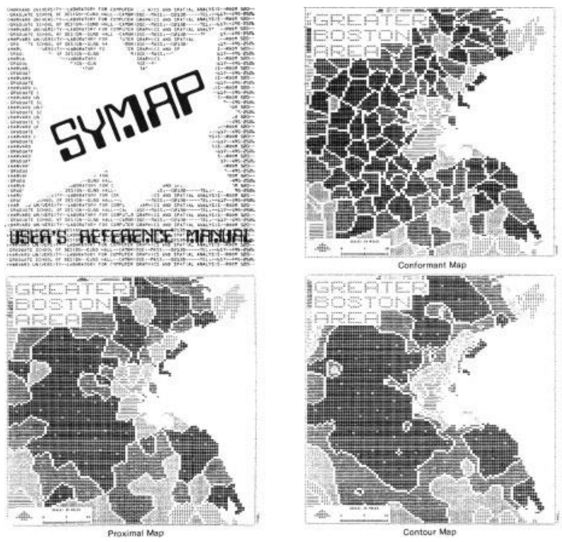

Figure 2.7. Symap Manual (ESRI Press, 2005)

Another research that leads to the development of GIS community was Tomlinson et al. (2014). Tomlinson is also known as the father of GIS after he created the first computerized GIS software in developing The Urban and Regional Information Systems Association (2013) for Canadian Land Inventory use of 1960. In 1968, he presented a paper titled "A GIS for Regional Planning" and has been proactive in helping GIS community. GIS is able to interpret different types of data and has the potential to display those varied data on one map. GIS is also applied in organizations, schools, governments, and businesses. This tool keeps all the data collections in a form of latitude and longitude, postal zip code, census tract name and so on. The map is the main product of GIS and it is used to display answers to queries. GIS is commonly linked to any type of applications and operations, which is business related, telecommunications, logistics, management, insurance and so on. Other than that, this system tends to locate the features on the earth's surface in the context of analyzing the geographical patterns. There are tons of map layers for the transportation networks, jurisdictions, economics and population (Fu, Sun, & Yin, 2011).

2.5.2 How do GIS Works

GIS stacks layers of information. Each layer represents a different type of information, such as agriculture yield, industrial area, forest area, settlement area.

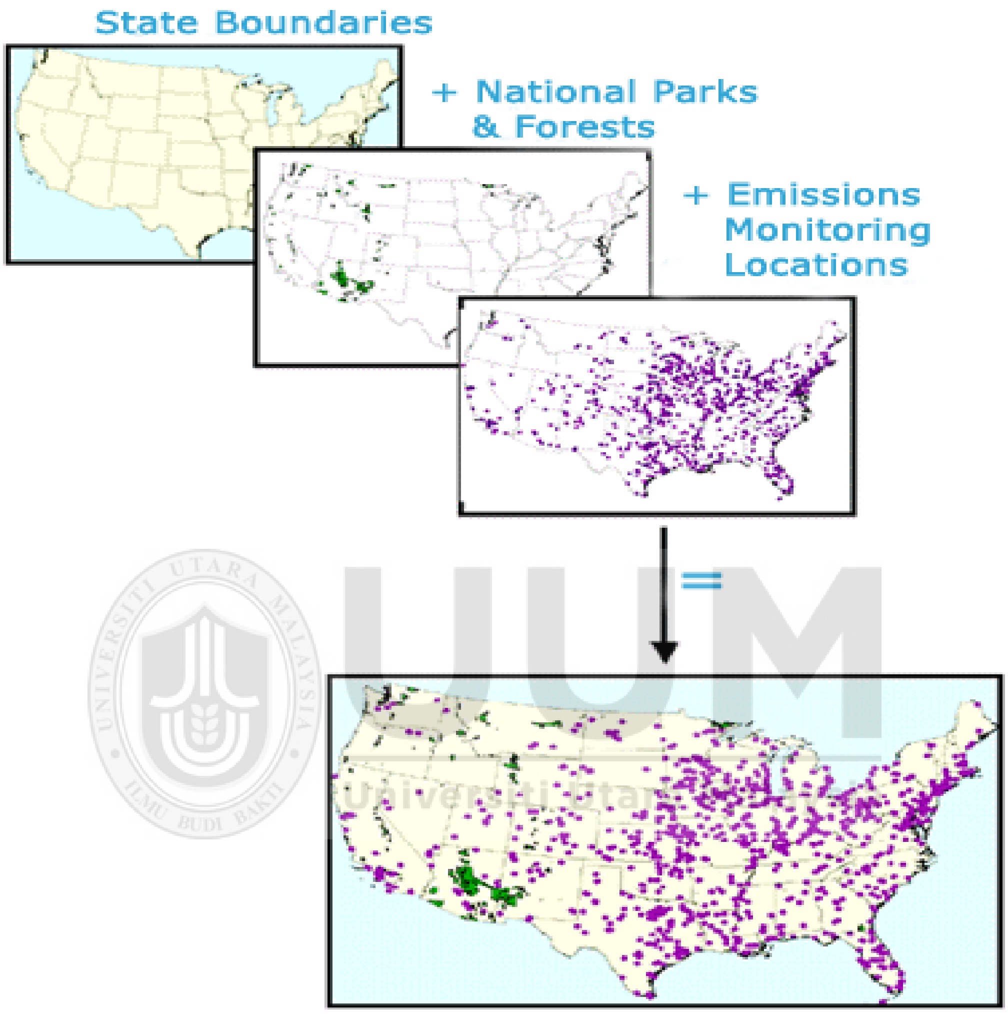

Figure 2.8. US Environmental Protection Agency on "How GIS Works" (EPA(2014))

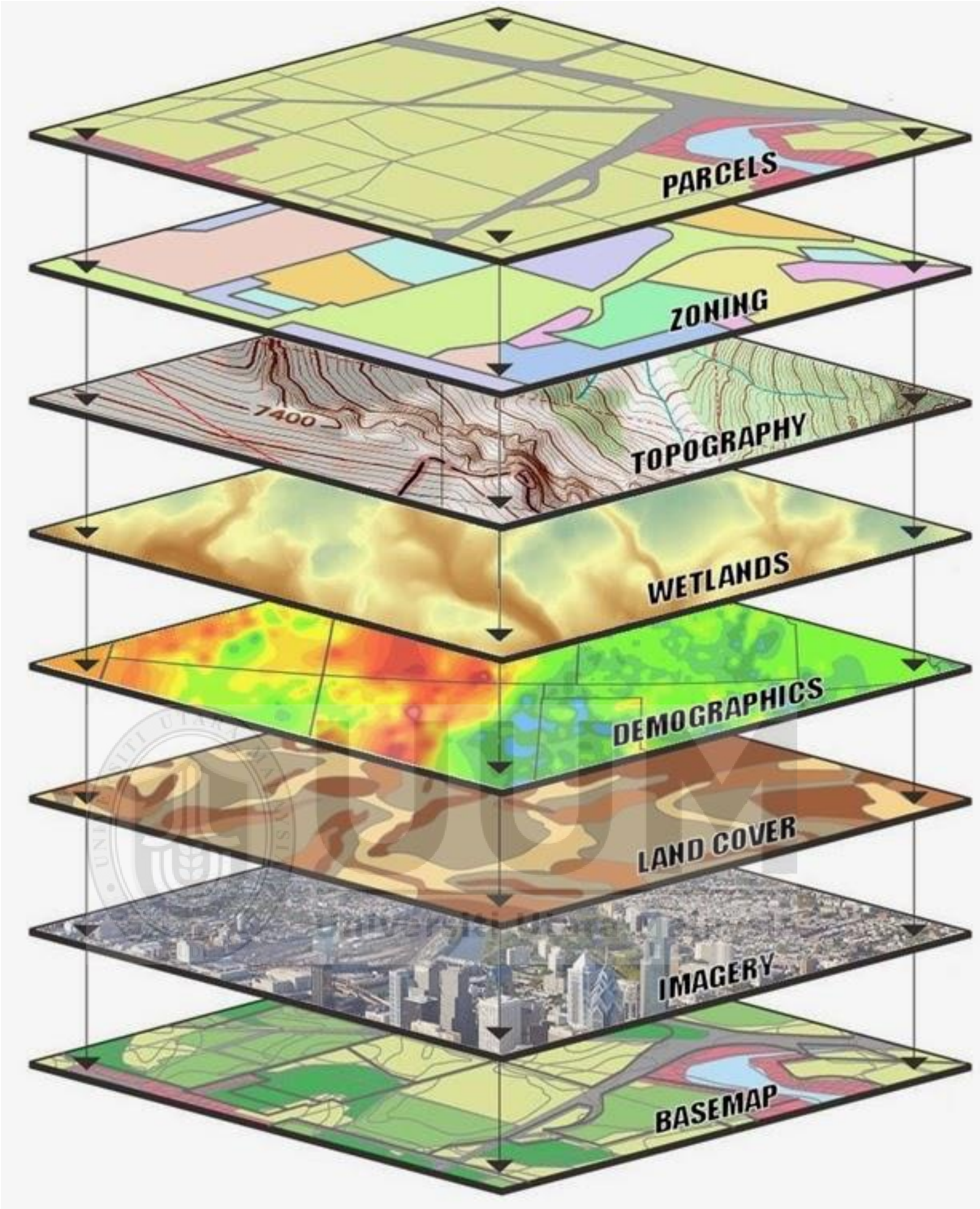

Figure 2.9. The Representation GIS Layer Which Contained Different Types Of Information. UDEQ (2015)

Figure 2.8 shows how US EPA use multiple layers of topographic images to project information like state boundary, a national park with forest and emission monitoring locations layered to show GIS information. While UDEQ (2015) show how information is layered out starting from the based map which shows only the terrain information, other information can be added onto the base map such as satellite imagery, area that have land, demographics of a population, wetlands, topographic map which contains a lot of information in 2D format, zoning of a state or country and parcel showing land that is owned by individual or company (refer Figure 2.9). Figure 2.8 and 2.9 presented information or specific data of an area overlaid with other data such as agricultural and forest area. With the capability to combine various data into layers which can show information based on layers, adds more information regarding the terrain data.

2.5.3 Process of Acquiring GIS Data

Numerous studies on retrieving GIS data have been conducted. The study by Baccini, Laporte, Goetz, Sun, and Dong (2008) generated African tropical area that uses electromagnetic spectrum to generate the vegetation area. Another is using aerial photography as can be seen from Morgan, Gergel, and Coops (2010), which examined the usage of aerial photography for ecosystem management for long periods. Makanga et al. (2015) utilized public GIS data and also from Mozambique National Cartography and Remote Sensing Centre (CENACARTA) to propose a framework of cost-efficiency on acquiring GIS data. Yildirim (2012) examined the suitable of solid waste management area in Trabzon Province, Turkey by generating raster data area using multiple maps and satellite data imageries. Crooks (2010) uses vector data to represent the residential area.

2.5.4 Types of GIS data

There are two main types of GIS data when retrieve, which are Vector and Raster. Vector data type is usually represented as points or dots also by connecting the dotted lines creates a line formation that forms a shape that is called polygon. Figure 2.10 shows how vector data works.

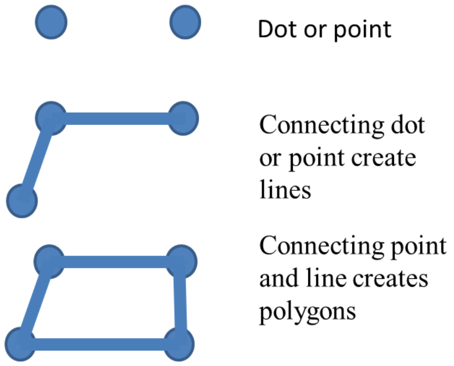

Figure 2.10. Vector Data Type

Coordinates in the vector are based on the position of X and Y of the dotted lines and polygons. For raster, the data is stored in grid or matrix. Figure 2.11 shows how raster data works in a grid or matrix form.

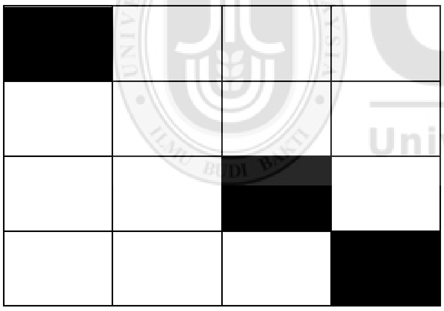

Figure 2.11. Raster Data Type

Raster data can be used in images as images apply pixels that use grid and matrix form. Raster data can be divided into two main types which are: discrete for data that is static, and continuous for dynamic data. Data that is static can be considered as discrete like mountains, hills, seas, while data that is dynamic and considered continuous are populations, rains, and flood, technological advancements allow these two data types to be converted to each other. i.e., raster can be converted to vector and vector can be converted to raster (GISGeography 2016) (The University of Washington Spatial Technology GIS and Remote Sensing, 2013).

2.5.5 Projections of GIS Data

Projections in GIS is applying world coordinates onto the GIS data that have been collected, there is two main projection for GIS data which is usually used which are Geographic coordinate systems (GCS) and Projected coordinate systems (PCS). The geographical coordinate system consists of degree unit of measure, a prime meridian, and a datum for globe or spherical referencing. That has the equator for zero latitude and prime meridian for zero longitudes, the datum is referring to mathematical correction of GCS data. PCS refers to projection on 2D surfaces; PCS allow calculation in metrics such as metres, kilometres and miles on 2D surfaces as the value of PCS is constant. Map projection, however, is a combination of GCS and PCS with a mathematical calculation to get the correct coordinate. This calculation is needed as map projection will have distortion as projecting spherical value onto 2D surfaces (sgerhardt, 2011) (ArcGIS Resource Center, 2012).

2.5.6 Application of GIS

GIS technology has been used in various fields and has benefited the community with information. In a study by Incekara (2012), GIS was used to show demographics of the student achievement, highlighting the important role GIS can play as a learning aid in geography by the teachers. In another research by Karakuyu (2010) also acknowledged that GIS is a powerful learning aid. A case study conducted by Superego showed that GIS can also be used as historical learning aids as it can show the mapped area of the previous historical cities of Taiwan (2006).

GIS is also used as an in disaster control management. The study of Xu *et al.*(2012) used GIS to measure danger level of landslides. Attaway et al. (2014) in his research looked at the potential areas of high dengue cases in Kenya, incorporating the GIS system to locate the potential areas with high cases of dengue. Chang *et al.* (2009) used GIS in a similar study as Vasiljević et al. (2012), which used GIS in to look at the land area that is suitable and unsuitable for waste disposal.

2.5.7 Mobile GIS

As technology advances, a new way of visualizing GIS application has been made through the use of mobile applications. Ismaeel and Hamead (2014) research on providing information on pregnant women position and status using google map and android system. Jajac, Stojanovic, Predic, and Rancic (2013) studied the efficiency of a specific task using mobile GIS data. Google Earth (2014) and Google Map (2014) both have the capability to run on both Android and iOS. Tsou (2004) studied the frameworks of a Mobile GIS and explained in details the infrastructure of Mobile GIS which can be used for monitoring environment and environmental management. Stenneth, Wolfson, Yu, and Xu (2011) studied the detecting transportation mode using GPS information. Mobile with GIS capability can also be used as an alert system using Volunteered Geographical Information (VGI) (Oxendine & Waters; (2014)).

2.6 Terrain Visualization Process

In recent years, a lot of new technologies have emerged and this contributes to different process regarding terrain visualization. Terrain visualization data or DEM data comprises of data in grid formats (Moore, Gessler, Nielsen, & Peterson, 1993) and data processing is needed to make sure the data is minimal on errors (Raber, Jensen, Schill, & Schuckman, 2002). Generally, there are two main techniques in visualizing terrains that are manual and automated. The manual technique of visualizing terrain involves method is mountain representation using molehill shape, another is skeletal lines that show information regarding mountain crests, ridgelines, and streams in skeletal shape lines (Ruzinoor et al., (2012)). Profile lines in a cross-section view of a terrain surface while Hachures is a technique that shows terrain surface in lines that can be close to shading where each line represents slopes of the terrains. Shading is using a darker shade of a single colour to show the contour of the terrain surface. Ruzinoor et al. (2012) divided automated techniques into two; that is photo realistic that attempt to generate the terrain using OpenGL and advanced algorithms with colour to differentiate the terrain information such as height map and contours also with overlaying high-resolution satellite imagery. Non-photo-realistic approaches use computer-generated silhouette shading to show the terrain surface view. Ruzinoor also focused on web-based 3D terrain visualization that uses a different process to generate the terrains such as, Virtual Reality Markup Language (VRML). This is further explained in Huirong Chen, Peng, Li, and Yu (2009), from their studies in generating 3D terrain using VRML. Ruzinoor et al. (2012) in another research compares it to three different processes that are Overlaid Satellite Image, Colour Shading, and Silhouette Rendering Algorithm. Veronesi and Hurni (2015) used relief shading with different light direction angle. Röhlig and Schumann (2016) studies on using occlusion to look at hidden areas of a terrain by using the widget that was developed. Another process is using GPU to visualize the terrain data. González, Pérez, and Orduña (2016) tested the performance of GPU on visualization terrain data. In a study by Dübel and Schumann (2017), developed a terrain visualization viewer in a high-end computer specification and studied the FPS of the viewer.

2.6.1 Terrain Visualization Process Using VRML

There have been several studies relates to terrain visualization using VRML. Ruzinoor, Abdul Rashid, Pradhan, Ahmad Rodzi, and Mohd Shafry (2013) studied the effectiveness of different GIS data with web VRML environment, and on the performance of the different GIS data types when visualizing in VRML web environment. The studies were about the FPS, Loading and response time, file size, memory and CPU usage of each different data type. Ruzinoor (2011) also applied the study but with three different web servers locations and hardware. Wang, Li, and Zhang (2014) created VRML terrain for robot movement simulation at sloppy areas of a terrain. X. Wang, Xuedong, Jiangfeng, and Dan (2012) uses VRML terrain to simulate a driving environment simulation.

2.6.2 Terrain Visualization Process Using HTML5

Terrain visualization has changed a lot in the past several years. Amongst these changes is the introduction of HTML5 for terrain visualization. Cellier, Gandoin, Chaine, Barbier-Accary, and Akkouche (2012) studied on reducing terrain size and visualizing terrain data using HTML5 through an algorithm to combine and reduce certain parts of the terrain data. Roccatello, Nozzi, and Rumor (2013) developed a framework that utilizes HTML5 application programming interface (API) with Web Graphics Library (WebGL) and also compliant with Open Geospatial Consortium (OGC) to view large-scale terrain data in a web environment. Latifoski, Kotevski, and Hristoski (2016) used HTML5 as well WebGL to simulate flooding in the large area. Bahor and Ramic-Brkic (2013) compared the performance of HTML5 and Adobe Flash by looking at its FPS as well as its memory and CPU usage.

2.6.3 Terrain Visualization Process Using Game Engines

In recent years a lot of new techniques for terrain visualization have emerged. Amongst these techniques, the game engine is a promising one. The game engine is a very powerful tool to create games and also applications that are beneficial for the society. Wang et al. (2010) study how Unity3D and GIS data can be utilized to create and plan for Henan Institute of Urban Construction campus development that allows the user to navigate and explore the environment created but did not mention the hardware used as well as the efficiency of the prototype. Humbert et al. (2011) simulated a 19^{th}-century town that includes the terrain information and satellite imagery that was converted in RAW format before converting into 3D modeller then inserted into Unity3D. This study too did not mention the performance of the prototype. In another study by Bishop (2012) on using torque game engine with AR technologies on smartphones to send information regarding the terrain. However, the author did not mention which model nor what hardware capabilities that are used to create the prototype. Yang et al. (2004) used torque game engine to visualize terrain of Tongariro National Park and also explained the performance of the prototype in CPU, RAM, and FPS. However, it did not mention the loading time and response time and also used same GIS data. Rathnam et al. (2009) and Prasithsangaree et al. (2003) use the Unreal engine to create a military simulation that uses real-world terrain data. However, the study did not explain the performance information of the development when using unreal engine. Kang, Kim, and Han (2015) used Ogre3D, Unity3D, and Open scene graph (OSG) terrain data to create a virtual environment for military planning. Figure 2.12 shows the three different results of terrain visualization that was developed by them. However, the study only focuses on visualizing the terrain in different software and is yet to show the performance of each software.

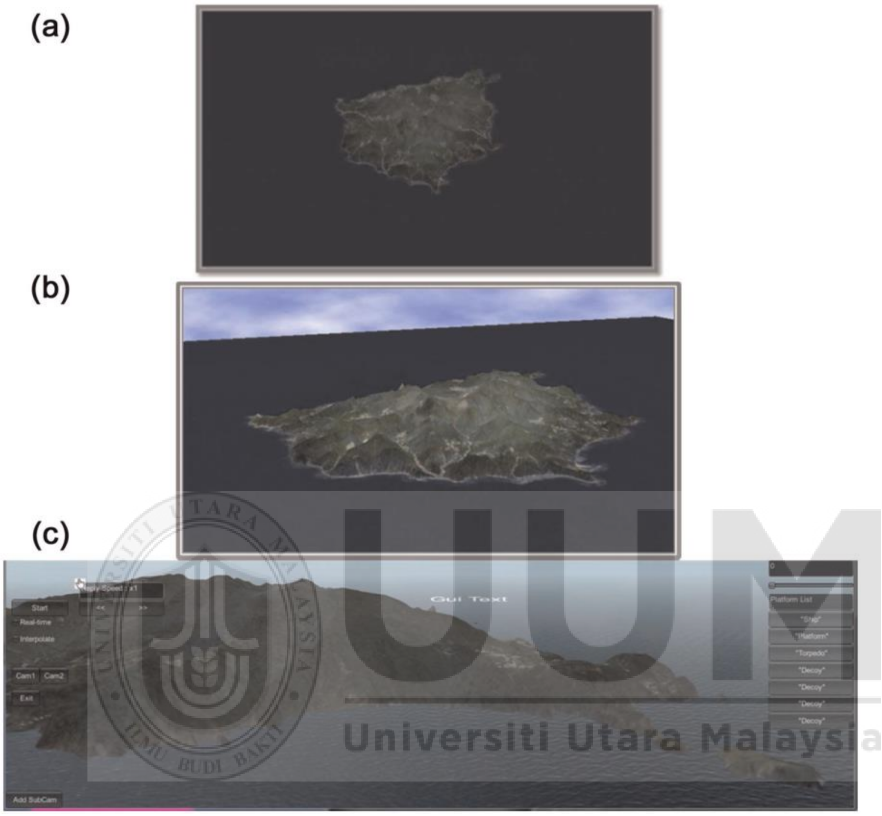

Figure 2.12. Results of Terrain Visualization *(Kang, Kim, & Han, 2015)*

Shin, Beirami, Cho, and Yu (2015) used Unity3D to visualize the terrain for sea navigation support by using raw terrain data, the studies utilize 3rd party paid plug-in to enable Unity3D to read terrain data. It too did not show the performance of the development result. Jarvis, Løvset, and Patel (2015) used Unity3D to create training simulation for industrial safety by using Unity3D terrain engine to simulate the environment of the study. They mention the GPU information but did not explain the performance of the prototype. Figure 2.13 shows the results of their study. GMV (2012) explained methods of visualizing DEM data inside Unity3D, one is using

3DEM (2014) and Terragen (2016) then convert into raw data. GMV also mention the process needed to display the DEM data. For method one, DEM data that was collected is then process for Terragen usage inside 3DEM, the area needed is then defined inside 3Dem. after defining the resolution that is needed, it is then exported for Terragen usage inside Unity3D. The terrain data is exported again into a raw format that later can be loaded into Unity3D terrain engine as RAW data.

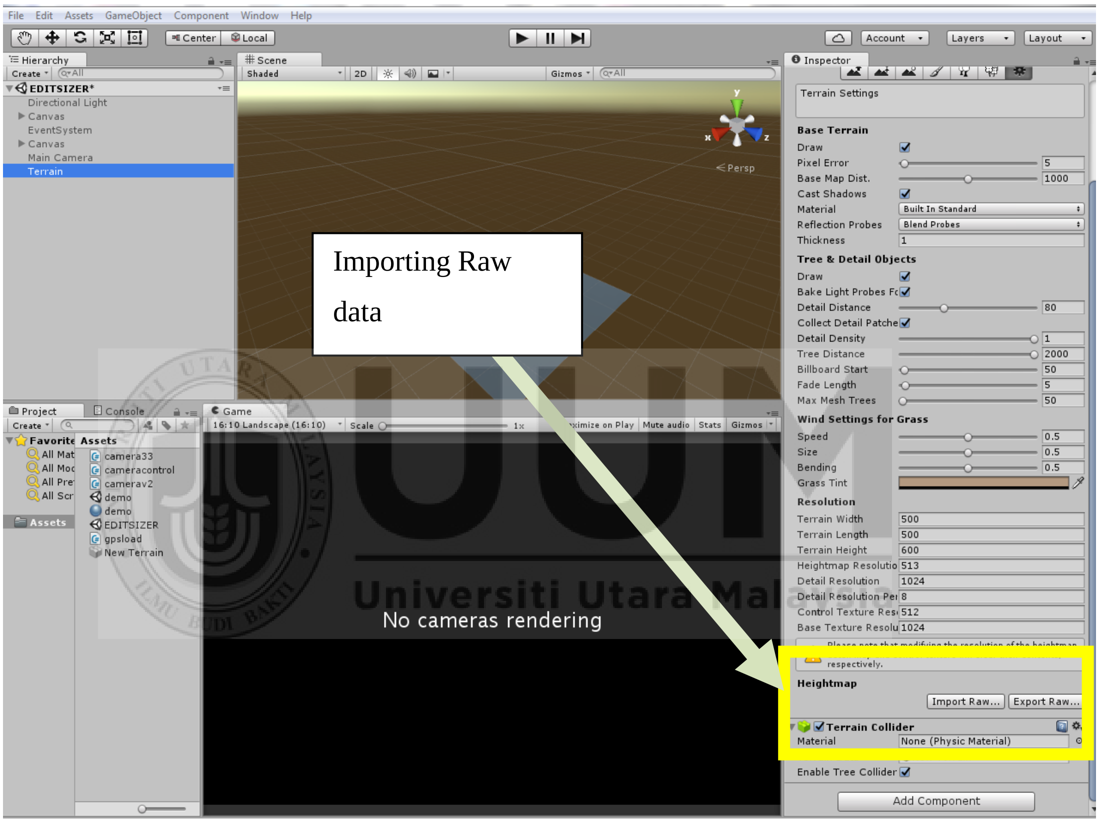

Figure 2.13. Importing Raw Data from Terragen.

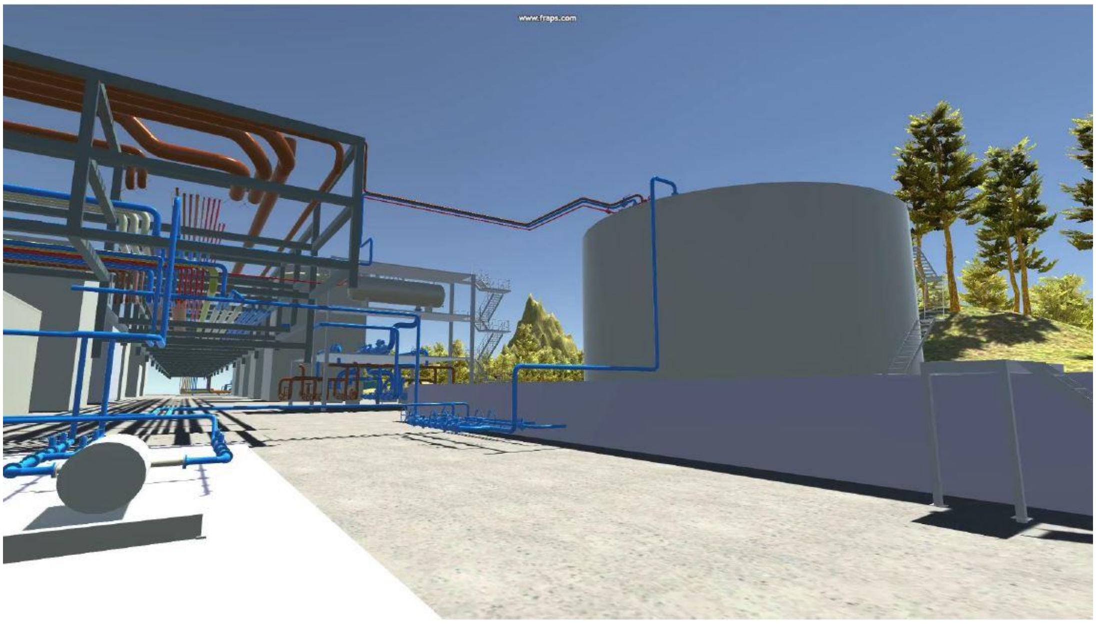

Figure 2.14. Terrain visualization that was created using Unity3D *(Jarvis, Løvset, & Patel, 2015)*

Table 2.2 shows the list of different technique from different types of game engine for visualizing terrain. The games used included Unity3D, UDK, and TGE. It's clearly shown that most of the game engine using their 3D modeller for importing the terrain data inside their engine.

UDK terrain visualization process is called landscape editor and uses a raw data format that is retrieved from image editing software that converts DEM data into 8 or 16bit grayscale RAW data format before it can be loaded into UDK. The process of visualization inside UDK uses vertices and Quad that is predefined by the user before the RAW data is imported into UDK, in applying the UAV imagery UDK need to use 8bit single channel RAW file that is loaded into the layer before it can be applied on the terrain that been generated (Epic Games, 2012).

CryEngine terrain visualization process is called world machine and it uses also 8 and 16bit grayscale RAW data format that has been converted from image processing software before it can be loaded by the world machine and converted again into format is that is readable by CryEngine, the same task have to be done before UAV image can be applied to the generated terrain (Endres, 2017). Both landscape editor from UDK and world machine from CryEngine requires the user to manually configure the terrain requirements inside the game engine terrain itself before the terrain visualization can be achieved. The process of generating the terrain data is much more complicated compared to an enhanced process that is applied in Unity3D as the user needs to select the terrain data that can be loaded inside Unity3D. Figure 2.15 shows the terrain view generated inside torque game engine while Figure 2.16 shows the terrain created using UDK.

Table 2.2

List of terrain visualization technique based on game engine

Articles Title	**Game engine**	**Technique Terrain visualization**	**Software's used for modelling**
A new method of virtual reality based on Unity3D	Unity3D	Uses 3D modeller then is imported inside Unity3D	3D s Max and AutoCAD
Use Of A Real-Time 3D Engine for The Visualization of a Town Scale Model	Unity3D	Uses 3D modeller then is imported inside Unity3D	Handyscan and Maya
A new method of virtual reality based on Unity3D	Unity3D	Uses 3D modeler then is imported inside Unity3D	Sketchup and AutoCAD

Table 2.2 cont.

On-Line Approaches to Data Delivery and Visualization in Landscape Planning and Management	TGE	Uses 3D modeller then is imported into TGE	spatial information exploration and visualization environment (SIEVE) and ArcGIS map
On-Line Approaches to Data Delivery and Visualization in Landscape Planning and Management	TGE	Uses 3D modeller then is imported into TGE	spatial information exploration and visualization environment (SIEVE) and ArcGIS map
Game Engine Support for Terrain Rendering in Architectural Design	TGE	Uses 3D modeller then is imported into TGE	AutoCAD and ArcGIS
Incorporating large scale SSRR scenarios into the high fidelity simulator USARSim	UDK	Uses 3D modeller then is imported into UDK	openscenegraph and ArcGIS
TSAF: a simulation bridge between OneSAF and the Unreal game engine	UDK	Uses 3D modeller then is imported into UDK	Open flight and 3D max

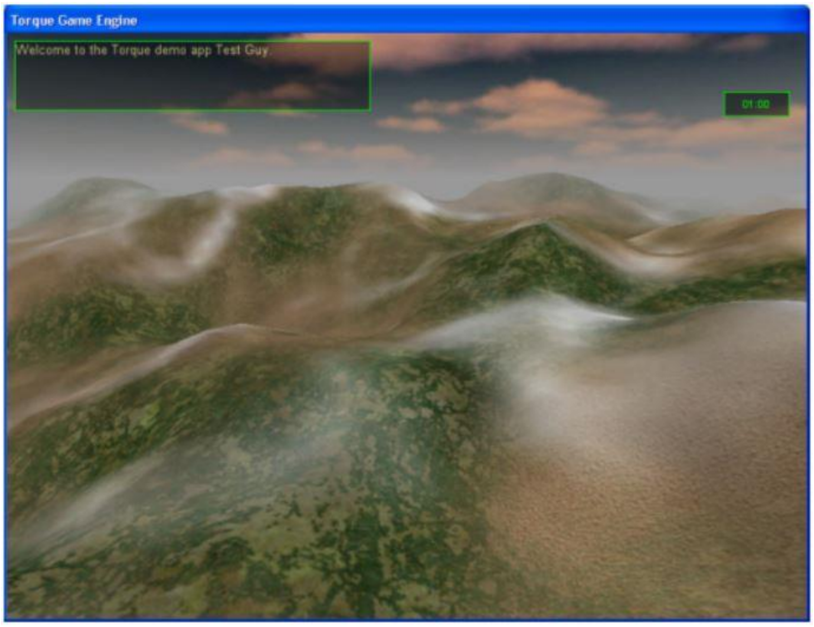

Figure 2.15. Terrain Visualization Generated from Torque Game Engine (Source: Yang, et al (2004))

Figure 2.16. Terrain Visualization Generated from UDK Game Engine (Source: Bishop (2012)

2.6.4 Performance of Terrain Visualization Process

Performance can be explained as a measurement to find out the efficiency of the task or tools used. In terms of terrain, visualizations are to look at how terrain data format, terrain data size, and software that uses the terrain data affect the performance of the published applications. Terrain visualization performance can look into two types of performance environments; that is online and offline. In an online environment's operating system that runs the terrain visualization application activate the operating system networks service to enable online data transaction to happen in online environments, while in offline environments the operating system the service enters into idle mode (Microsoft, 2017). Performance in visualization terrain that can be measured while in online and offline environments is different. In online environment loading time and response time is added to the measurements.

2.6.4.1 Loading time

Loading time is the time needed to fully load the webpage onto the user's computer. Wang, Liu, Guo, and Chen (2014) explained that loading time is an interval from the start of the request until the finish of the request of a webpage. In their research, the performance of two different web pages by reusing similar layout on the different webpage was studied.

2.6.4.2 Response time

Response time also as mentioned previously is the time the first user sent it to request to the webpage to access and load the webpage. Both Loading time and response time will be measured in milliseconds or (MS). Natarajan, Amer, and Stewart (2008) studied on improving the response time of a webpage by using pipelining that is available as a plug-in or built-in by browsers such as Firefox. The studies of Jacob (2016) and Low (2017) mentioned that incompatibility with the plug-in will result in slow response time and loading time. While in offline environments these resources are used to measure performances in terrain visualizations.

2.6.4.3 Frame per Second (FPS)

Fps is how many frames are shown in one second. Frames can be seeing a single image and how many images are going to be shown in that exact seconds. Fps will be measured either in 30 or 60 frames closest to each. Pranckevicius (2014), and Bose and Rajagopala (2012) explained how FPS can be improved by allowing custom configuration. Cheung, Kanade, Bouguet, and Holler (2000) studied the performance of FPS using 3D voxel with a human motion while experimenting human motion in 3D space. A slow FPS would result in a delay in input and reactions.

2.6.4.4 Memory Usage

Memory usage is how much an application uses the user computer memory basically known as Random Access memory (RAM). The measurements will be in bytes. A higher usage of RAM results in a slower computer performance itself and the computer will switch it's usage into hard disk as RAM and shows that the memory usage is not properly processed by the RAM (Hoffman, 2014).

2.6.4.5 CPU usage

The CPU usage is referring to how much your computer processor is being used when the terrain data is opened the data is shown in percentage. The computer with higher CPU usage won’t be able to work properly cause the performance of the computer will be slowed down (Smith, 2014).

2.7 Terrain Visualization Software

The emerging requirement of 3D terrain visualization has created a good number of software and terms regarding 3D terrain. Terms such as DEM, TIN, and DTM is normally used for terrain data type and all these terms are frequently used in software that has the capability to read and display 3D terrains. Ruzinoor *et al.*(2012) have reviewed multiple techniques and software for 3D terrain visualization of GIS data. The VRML technology has been used by Ruzinoor et al. (2008) to visualize terrain online. Horne (2014) the original author of 3DEM created the software for the commercial market and after a few years of development, he halted the project but still shared the full version of the software for public use. The software has the capability of importing and generating 3D terrain from DEM data. This software also provides a surface view of Mars terrains. Zuse-Institut Berlin and Lenné3D (2007) created the BioSphere3D. This software has the capability to display the landscape of terrain in the globe by using a data like the DEM, satellite and aerial images, 3D models (Collada), 3D plant models (Lenné3D (2014) Flora3D (2014) and Shapefiles. Cesium is a 3D terrain visualizer that uses globe view concept which has the capability to zoom and explore the terrain of the whole world. It uses WebGL libraries to load the Shapefiles and makeup of the terrain. It is based on an entire cloud and is fully dependent on scripting (Analytical Graphics Inc., 2014). Besides that, Earth3D is developed by Andre (2014) which uses Java and C++ as core programming code. The data for this software is generated from NASA, USGS, the CIA and the city of Osnabrück. GenesisIV is a landscape software created by Geomantics (Geomantics, 2014). The advantage of this software is the capability of generating realistic photos of the terrains. Hftool is introduced by Stock (1998) which is a Unix-based terrain viewer and the output from this software can be converted into different types of file formats. Integrated Data Viewer (IDV) is created by Unidata (2014) which is based on java programming. The software is able to create an animation of satellite images, viewing a grid view of terrain plot and also displaying a data based on the globe. LandSerf a terrain visualization software created by Wood (2014) is also a java based terrain visualization. It can handle multiple files in GIS format and also have a huge interactivity visualization. This software is also able to import and export raster format and capable of integrating with Garmin GPS receiver. Guth (2014) has introduced MicroDEM which is a portable terrain viewer where it can view DEM data and also able to merge the data. Virtual Globe was created by Sintef a research organization in Scandinavia that covers a lot of research area until it was taken over by Norkart for further development, this software uses globe as its main view and uses data such as SRTM, GLOBE, and GTOPO30 to display terrain model and this software is also developed using Java (Norkart, 2014). OpenEV was originally developed by Atlantis Scientific Inc. (OpenEV, 2014), which is currently known as Vexcel. As of 2006, it was acquired by Microsoft and has ceased involvement in the project. OpenEV is a terrain viewer that uses OpenGL for rendering and it supports various types of raster and vector format to create elevation. QGIS (2004) was created by Gary Sherman as a Git project in early 2004, since then QGIS have progressed and have a community of its own and it provides support to many GIS data format as well to multiple plug-ins providers by 3rd party developer can be added into QGIS. gvSIG (2015) is also a terrain visualization software that is capable doing 3D rending of terrain data. It developed as a community development by gvSIG Association which also developed a Mobile version of its GIS software. Another GIS software is named ArcGIS (2015) created by Environmental Systems Research Institute (ESRI), which allow support of multiple formats of GIS data and have a huge database of GIS data available using its cloud server, ArcGIS provides a lot of functionalities in manipulating GIS data with it ArcToolbox.

2.8 Theories Related To This Study

There are several theories that have been implemented to enhance the robustness of the proposed solution for terrain visualization. Animate vision theory seems to be suitable and related to this study. HIPO theory is also suitable to be applied for explaining the enhancing process involved in this study. The following sub-section discusses this theory.

2.8.1 Animate Vision Theory

Animations are among the other multimedia elements that comprise of text, graphics, audio, and video. Animations are a tremendous technique for enhancing visual impact to the representation (Rodgers, 2014). Animation comprised of interactive effects that allow users to engage with the presentation using devices like mouse or keyboard. Animated apparatus are common in mobile or desktop applications. Animations are categorized into two and three-dimensional animations. This technique creates the illusion of movements for the positions of models or puppets. Animate vision relates to the visual perception to transform the physical environment. Other than that, the illusion of movement is the perseverance of vision. The persistence of vision assists animators to develop smooth and really look like animations in a well-organized manner by only representing the frames to create the sense of motion. Experts claimed that animate vision provides the opportunity to view more natural visualization environment of virtual information (Santos, Chen, Taketomi, Yamamoto, Miyazaki, & Kato, 2014). Other than that, VR allows the user to travel inside the virtual world.

Animate vision theory was utilized to represent the three-dimensional terrain visualization in more detail and precisely from various perspective. Previously, most research studies regarding terrains utilized 2D terrain that provides shading and lines to show the terrains' information. Currently, the advancement of technology enables to view the 3D terrain visualization through the use of VR which can generate a lot of terrain information. Moreover, Unity 3D eases the process to the built-in terrain in-game engine that enables to create 3D terrain and build set to publish as standalone, Android and web application. In this study, 3D terrain visualization is proposed for the online environment because it provides a way for students, researchers, historian, geologist and many others to view terrain information details in 3D formats anywhere around the world as it is in an online environment. This study also utilized the game engine to allow the simulation of events, the researcher may use this to view dangerous area or simulations, and historian can use the terrain visualizations as a means to view ancient area in a simulated environment using real terrain data.

2.8.2 HIPO Tools

HIPO is a tool in software engineering that explains the structure of a system or documenting processes in a hierarchy form (TutorialsPoint, 2017). In a study by Anroni and Andest (2017) in developing an automated authorization of joint trading letter system using visual studio 2010, explained the process of the system in the

HIPO forms. Figure 2.17 shows the HIPO that was used in the study. The system is divided into three main processes which are inputs, process, and reports. The inputs are variables that would be used for the process and finally the report generating all in automated processed.

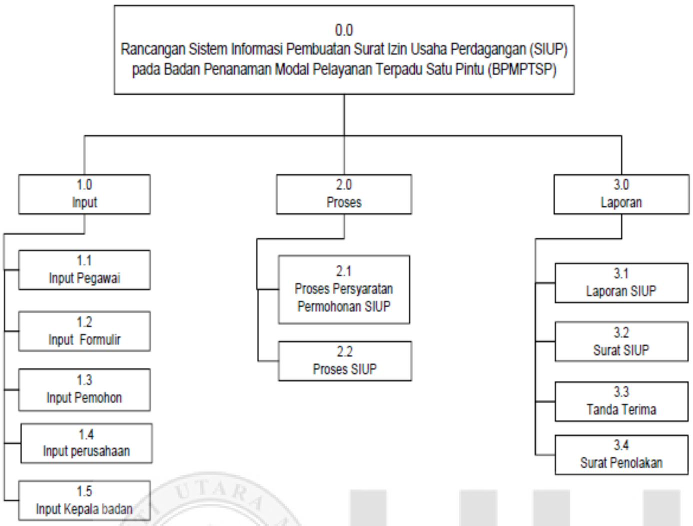

Figure 2.17. HIPO of Automated Authorization of Joint Trading Letter System.

In another study by Ningsih (2017) also used HIPO to explain the structure of his proposed system that is automated authorization of car park management. Figure 2.18 shows the structure of his system in HIPO model. The system processes are divided into three main phase that is entry, process, and report.

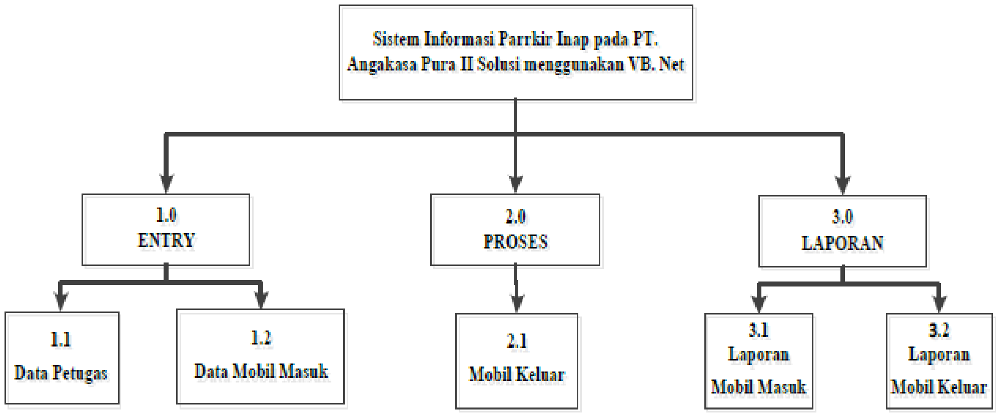

Figure 2.18. HIPO of Automated Authorization of Car Park Management.

Both studies that utilized HIPO explained the system structure and its subprocess. HIPO works in giving an overview of the system structure for documentation.

2.9 Summary

This chapter discussed the issues regarding visualization, virtual reality, game engine, GIS, terrain visualization process, terrain visualization software and theories related to this study. This chapter also briefly explains the history of game engines, architectures and how game engines are used and its contribution. A brief history of GIS and its applications like in mobile GIS was also discussed. The terrain visualization process which is the main part of this study is discussed in details relating to how it's being implemented on the various platforms such as VRML, HTML 5 and game engine. The main focus was on how terrain visualization process is used in the various game engines such as Torque, UDK, CryEngine, and Unity3D. All of the studies briefly mentioned about the performance of using terrain data inside their engine but the majority is not capable of visualizing terrain in an online environment. This study intends to enhance the 3D terrain visualization process using game engine mainly Unity3D and evaluates the performance of visualizing the 3D terrains.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

This chapter discussed the steps that are involved in achieving the objectives of this study. The methodology of this research was adapted from Offerman et al. (2009) design research process which contains three main phases which are problem identification, solution design, and evaluation. Figure 3.1 shows the relation of this studies as proffered by Offerman (2009) design research methodology. In the first phase, the requirements on how to enhance 3D visualization processes inside the game engine generate online 3D terrain visualization are gathered and analyzed for this study. Then, after all of the requirements were identified, the prototype was developed based on enhanced 3D visualization process all of these requirements and solution. Finally, the evaluations of the prototype compared with online and offline environments were made and the result was documented.

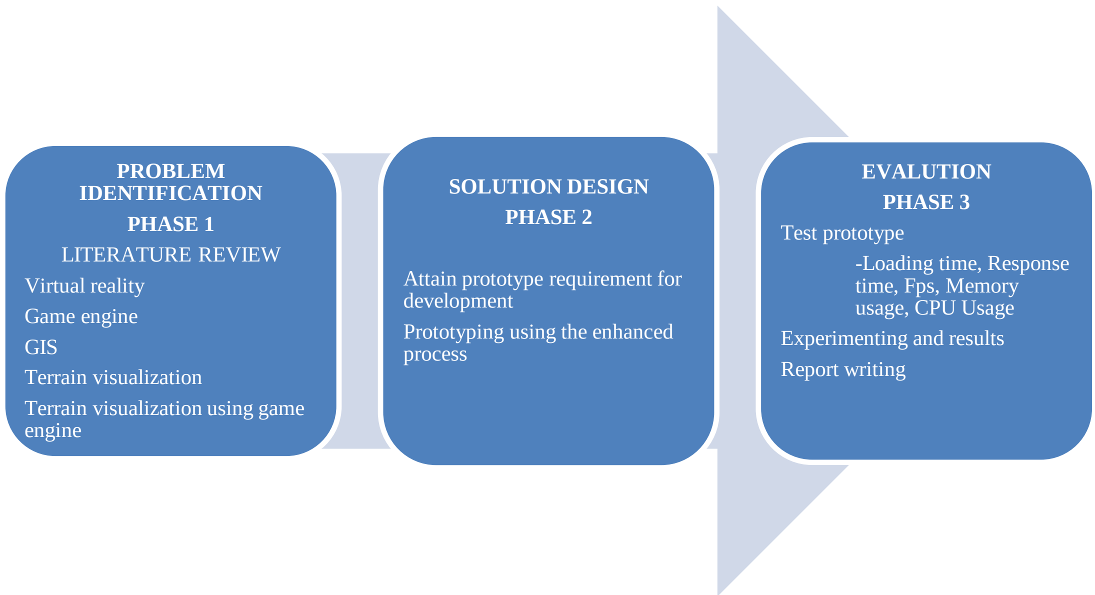

Figure 3.1. Design Research Methodology (Adapted from Offerman (2009))

3.2 Problem Identification

At this stage, the requirements of how to enhance 3D terrain visualization process inside the game engine for generating online 3D terrain visualization are gathered and identified. All of the requirements already discussed and reviewed in Chapter Two such highlighted how VR, game engine and GIS can be used together. The discussion on how each of the game engines could be used in visualizing 3D terrain and its limitations of each game engine is also discussed. It has been identified that the process of 3D visualization needs to be enhanced in an online environment to enable the 3D terrain to be visualized together with the real-world data (GIS data).

3.3 Solution Design

At this phase, the prototype is developed based on all of the identified requirements from the problem identification phase. The solution design phase follows the theory of Hierarchy Plus Input-Process-Output (HIPO) by Davis (1998) discussed on the implementation of the enhanced 3D terrain visualization process (refer Chapter 4). The prototype was developed by using Unity3D game engines. The focus is how t0 enhance 3D terrain visualization process applied inside the game engine. The enhanced process can generate the real 3D terrain visualization together with GIS data with online capabilities. The attain requirements and prototyping related to Unity3D were discussed in the next two sections.

3.3.1 Attain Requirements

Requirements are gathered from the literature review and experimenting with different types of software like ArcGIS and Unity3D game engine to understand the requirements needed for enhancing 3D terrain visualization process. The requirement is explored from three perspectives which are the process, software, and hardware. The Unity3D game engine software was used as the main platform. The web server was used to publish the prototype of the system for testing. Plug-in is required in order to run the prototype (e.g. Unity web player plug-in). The requirements of the hardware used for developing the prototype are a Desktop PC with 3.1 Gigahertz (GHz) CPU, RAM 8GB, GPU with 2GB built-in RAM, internet bandwidth capabilities (e.g. 512kb). The tools for measuring the performance of the prototype were also identified which include file size viewer, Page Speed Monitor, Firefox performance test and Process Explorer.

3.3.2 Prototyping

The Unity3D game engine was used for developing the prototype of online 3D terrain visualization. Unity3D is a cross-platform game engine that uses visual editor and scripting for customization. It is also able to create beautiful terrains, import 2D, and 3D model, and it is able to be published on multi-platforms such UNIX, windows and also mobile. The prototype can be divided into two phase which is design and development. The design phase begins with collecting the entire requirement needed for developing the prototype based on enhancing 3D terrain visualization process. Then, the development phase saw the application of the proposed design. Unity3D was utilized to run custom scripts that would run the process for displaying terrain in 3D. New terrain generated as an empty data without elevation data. Based on the experiment, it was found that the elevation data need to be in HDR and FLT in order to generate the real world terrain data inside Unity3D. The terrain generated can be viewed as a 3D object that has X, Y, and Z axis. The prototype can be viewed using free camera movement to view the terrains. The users can visualize and see the details of the terrain information in 3D environments. The prototype was published in Unity web player format so it can be accessed anywhere online as long as the user has internet access and also the plug-in viewer. The plug-in required for running the prototype is Unity3D web player plugin. Refer to Chapter Four for the details of the development process.

3.4 Evaluation

In this phase, the performance of the prototype based on enhanced 3D terrain visualization process in generating online 3D terrain visualization was examined. The method of collecting data used a quantitative approach which is measuring the performance of the process that generated the terrain inside Unity3D by comparing it with two different environments that are online and offline. Testing was done while the terrain is generated in an online environment as the well offline environment. In this phase, to verify the data collected from the prototype, the design of experiment was used as measurement tools to find out the comparison of each experiment that was conducted. The experiment was adapted from Sherif and Abdul-Kader (2011) experiments that studied frame rate, upload time, and visualization time. The experiment was focused on few criteria discussed below.

3.4.1 Loading time

Loading time is the first criterion being measured from the prototype. It was measured starting from when the user presses the "Enter" button or click search until the whole page has been fully loaded. It was measured in millisecond (ms). The prototype was published in the online and offline environment. Firefox add-on name "Page Speed Monitor" (2014) created by Fabasoft for Mozilla Firefox was used to measure the loading time. A study conducted by SEOchat (2017) was used as a comparison. Figure 3.2 shows how loading time is captured.

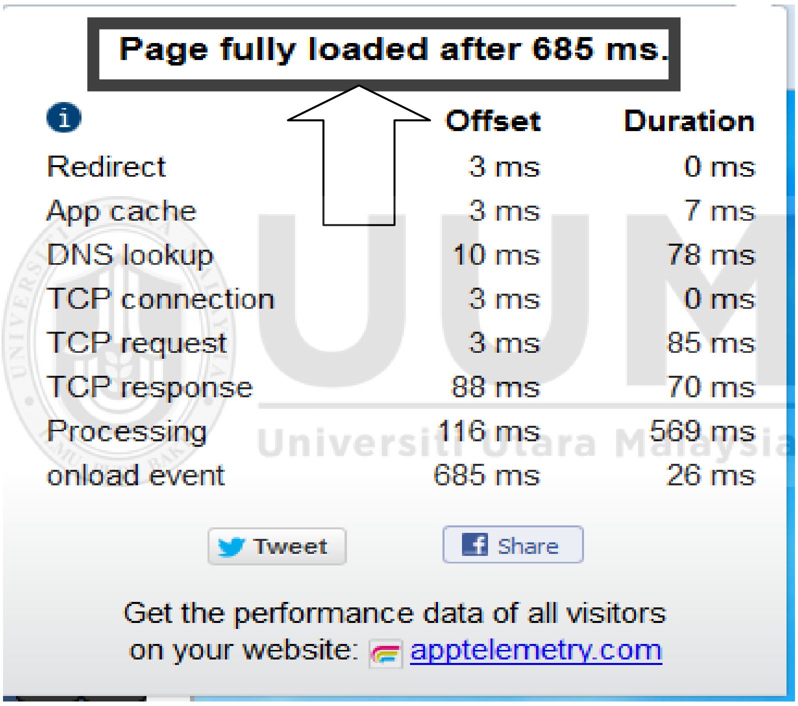

Figure 3.2. Page Speed Monitor for Measuring Loading Time.

3.4.2 Response time

Response time is the second measurement that was measured from the web-based 3D terrain visualization on the online and offline environment. Page speed monitor was also used to view response time. The response time was also measured in milliseconds (ms). The response time typically means the time taken from the first byte of page requests sent until the last received byte of the server response. This part of the study examined the response time of the prototype. Studies by Hussain, Wang, Toure, and Diop (2013) were used to compare the results. Figure 3.3 shows how response time is captured.

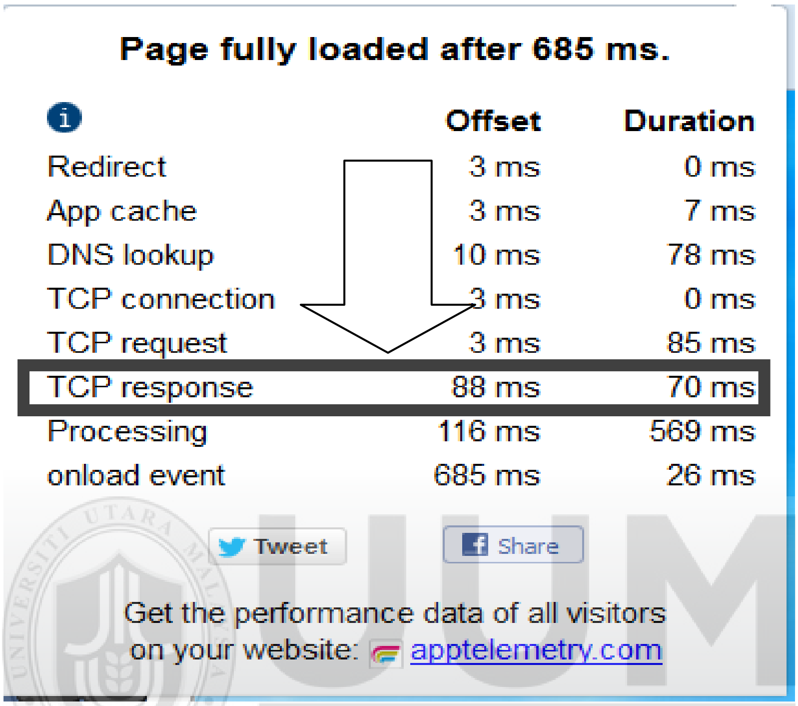

Figure 3.3. Page Speed Monitor for Measuring Response Time.

3.4.3 Frame per Second (FPS)

FPS is the third measurement that was measured while running the prototype. FPS can be defined as the measurement of how much information is used to store and display the motion. Each frame contained a single still image when the frames are executed consecutively it gives the illusion of motions. A higher FPS represents much smoother visualization and the lower FPS value represents nonsmooth visualization. Firefox performance tools are used to measure the FPS value of the prototype. This part of the study measured the frame rates of the prototype in web environments in the online and offline environment. Janzen and Teather (2014) were used to compare acceptable FPS. Figure 3.4 shows the meaning of frame rates or FPS, its shows each frame of the picture contains still images when 60 FPS referring to 60 still images that run in one second.

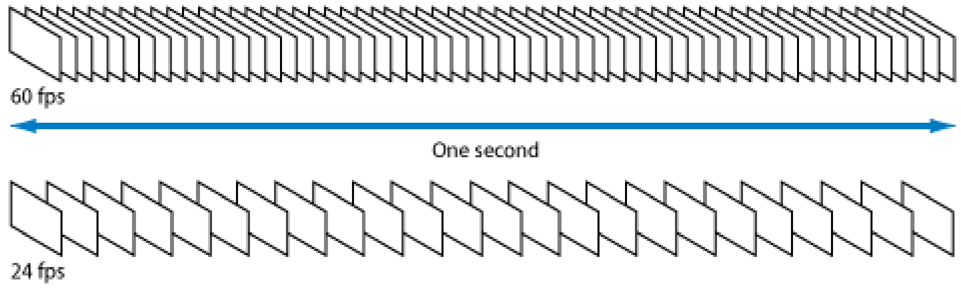

Figure 3.4. What is Frame Rate? (Apple Inc.,(2010))

3.4.4 Memory usage

Memory usage is the fourth measurement that was measured while running the prototype. This measurement looked at the memory usage of the prototype while running with different sizes of terrain data in the online and offline environment. This measurement used Process Explorer software which showed a clear view of the terrain cache inside the RAM during the running of the prototype. The study of Indraprastha and Shinozaki (2009) will be used as a comparison.

3.4.5 Data size

Data size will also be compared, being the fifth measurement that is to be measured while running the prototype. This measurement looked at the data size of the prototype while running with the different sizes of terrain data in the online and offline environment. A higher data size of usually will result slows in computer performance.

3.4.6 CPU usage

CPU usage is the fifth measurement that was measured by running the prototype. This measurement was looked into CPU usage of the prototype while running with different size of terrain data in the online and offline environment. CPU is known as the core of every computer. If a lot of processes run simultaneously at the same time, it will affect the computer performance. Process Explorer software (Microsoft, 2014) from sysinternal created by Microsoft is used for measuring the CPU usage while running the prototype. This software has the capability of showing the current CPU usage of each process that is running on a windows system. Figure 3.5 shows the Process Explorer software user interfaces.

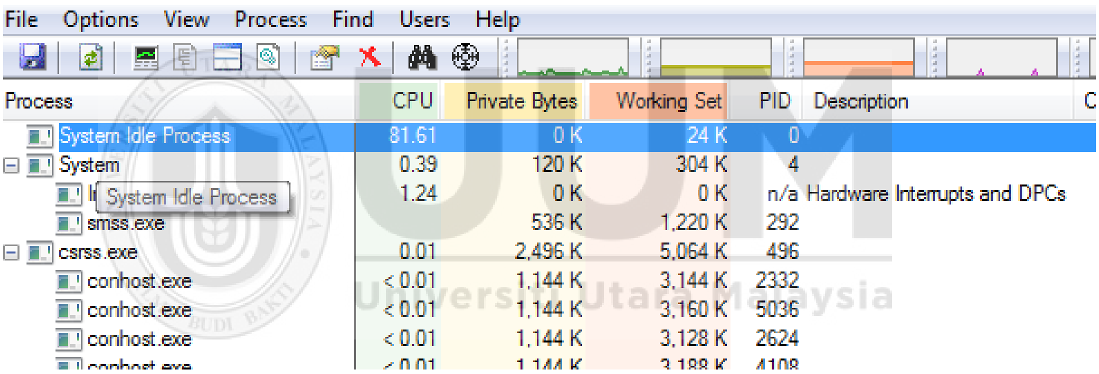

Figure 3.5. Process Explorer User Interfaces.

3.4.7 Comparison of all measurement for each terrain data size in Unity3D

The final measurement is to compare mainly all of the measurement before which are loading time, response time, FPS, memory usage and CPU usage in Unity3D on a single server (www.hasfaruz.net). But this time for loading time, the measurement was tested at two different times which are during office hours (in) and after office hours (out). The measurement was tested on different terrain data size as well. The results will be gathered and shown in one table which displays the whole value. The results were compared and analyzed.

3.5 Summary

This chapter discussed the overall methods and processes to be implemented in this study. This chapter explained the information regarding the tools and methods that were used for analyzing the results of the performance. The results that were measured included loading time, response time, FPS, memory usage, CPU usage and comparison of all measurement for each terrain in a single server. The results will be analyzed and measured into different terrain data size while running the prototype of web-based 3D terrain visualization in the offline and online environment. The results will be explained in more details in Chapter four and Chapter five. This is the proposed methodology to examine the performance of enhancing 3D terrain visualization process for generating terrain visualization using the game engine in offline and online.

CHAPTER FOUR
ENHANCED 3D TERRAIN VISUALIZATION PROCESS USING GAME ENGINE

4.1 Introduction

Terrain visualization is a technique that uses GIS data and visualizes it in a 3D object with their elevation data. Within this chapter, objectives one and objectives two will be explained in this chapter. This chapter explained on how to visualize the terrain by using the enhanced process for the Unity3D game engine based on enhancing 3D visualization process and its performance in online and offline environments. Starting from the data being captured until performance testing was explained in this chapter. The information gathered from this study can be used as a guideline for the developer in order to build their own terrain visualization system. The process begins with data capturing using UAV in a palm oil plantation in Sintok, Kedah then the data being processed for generating the metadata of the terrain in DTM and DSM. After that, the data was ready to be imported intoUnity3D for generating terrain in the 3D environment. While inside the Unity3D, the data being loaded into terrain engine and 3D model of the terrain were generated. With Unity3D, the player control and other features can be easily set before publishing into the web environment. Further detail on development and testing were explained in this chapter. Figure 4.1 shows the development workflow of web-based terrain visualization using the Unity3D game engine with enhanced 3D visualization process.

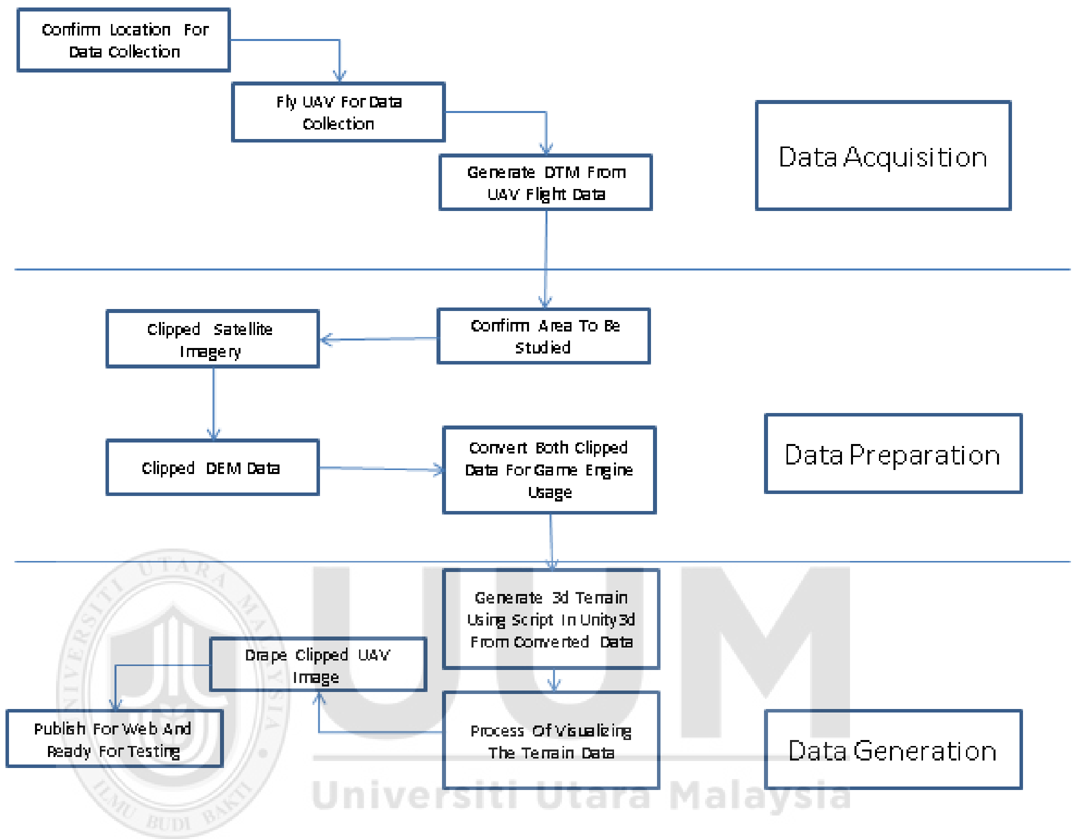

Figure 4.1. The Development of Online And Offline Terrain Visualization Workflow Using the Unity3D Game Engine with Enhancing 3D Visualization Process.

4.2 Data Acquisition

There are several ways of acquiring the terrain data with GIS coordinate as mention in Chapter Two. This study uses the data that was captured by using UAV. The study area for this study was Sintok Oil Palm Plantation near the Universiti Utara Malaysia. During data capturing a very high definition camera which has high resolution was attached to it. The data captured from UAV during flight is in high definition images (RGB), and the terrain data (DTM) together with its coordinate.

After all of the data has been collected, it needs to be further processed to generate the detailed accurate data.

Raw Terrains data that was acquired from UAV contain 100 images which needed to be stitch and the image does not have any georeferencing information attach to it. Ground Control Point was needed in creating georeferencing for the image. The UAV uses autopilot to fly on a set path. Figure 4.2 to 4.4 show the UAV, UAV flight path, and Captured images set.

Figure 4.2. UAV used for Acquiring the Data for This Study.

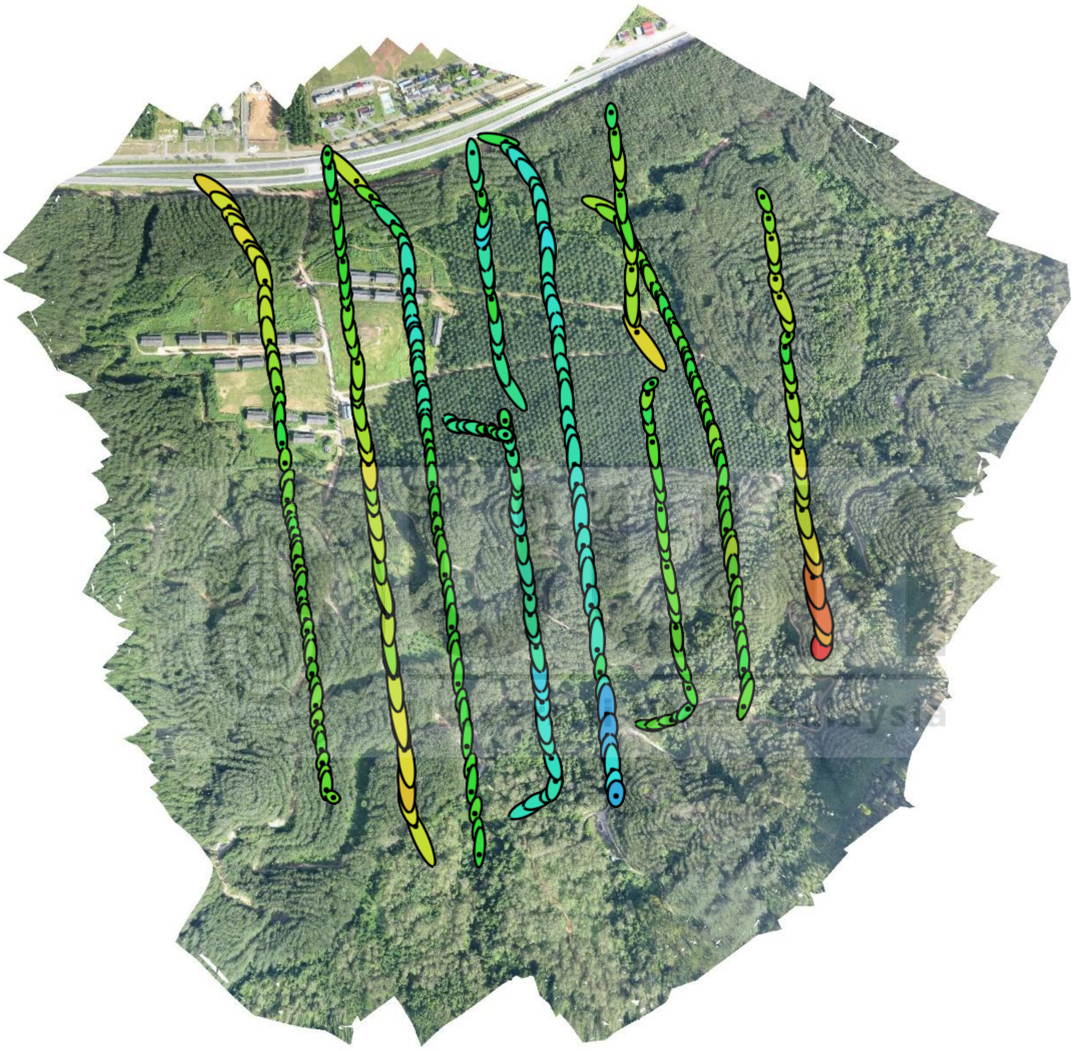

Figure 4.3. The Flight Path of the UAV

Figure 4.4. Area Captured by the UAV.

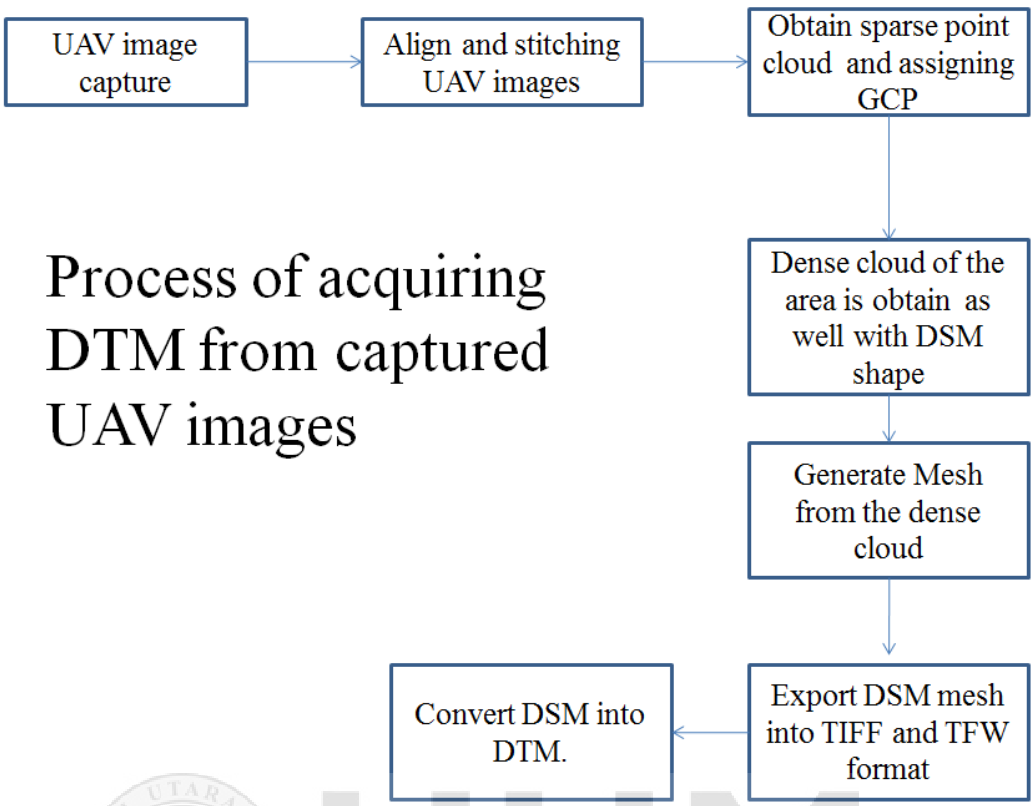

Figure 4.5. The Process of Acquiring the DEM Data from Captured Images Using Agisoft Photo Scan.

Figure 4.5 explain roughly how DTM data is acquired from UAV images. The process uses Agisoft photo scan (2015). The process starts by stitching the images that were captured by the UAV. After stitching sparse cloud is generated and Ground Control Point (GCP) is assigned to the sparse cloud, the Dense cloud is generated after the sparse point has obtained enough GCP, DSM can be obtained from the dense cloud. The next process is exporting the DSM into TIFF format. The DSM is converted into DTM by using DSM2DTM a feature from Geomatica (2015) software. Figure 4.6 shows the DEM generated from Agisoft photo scans software.

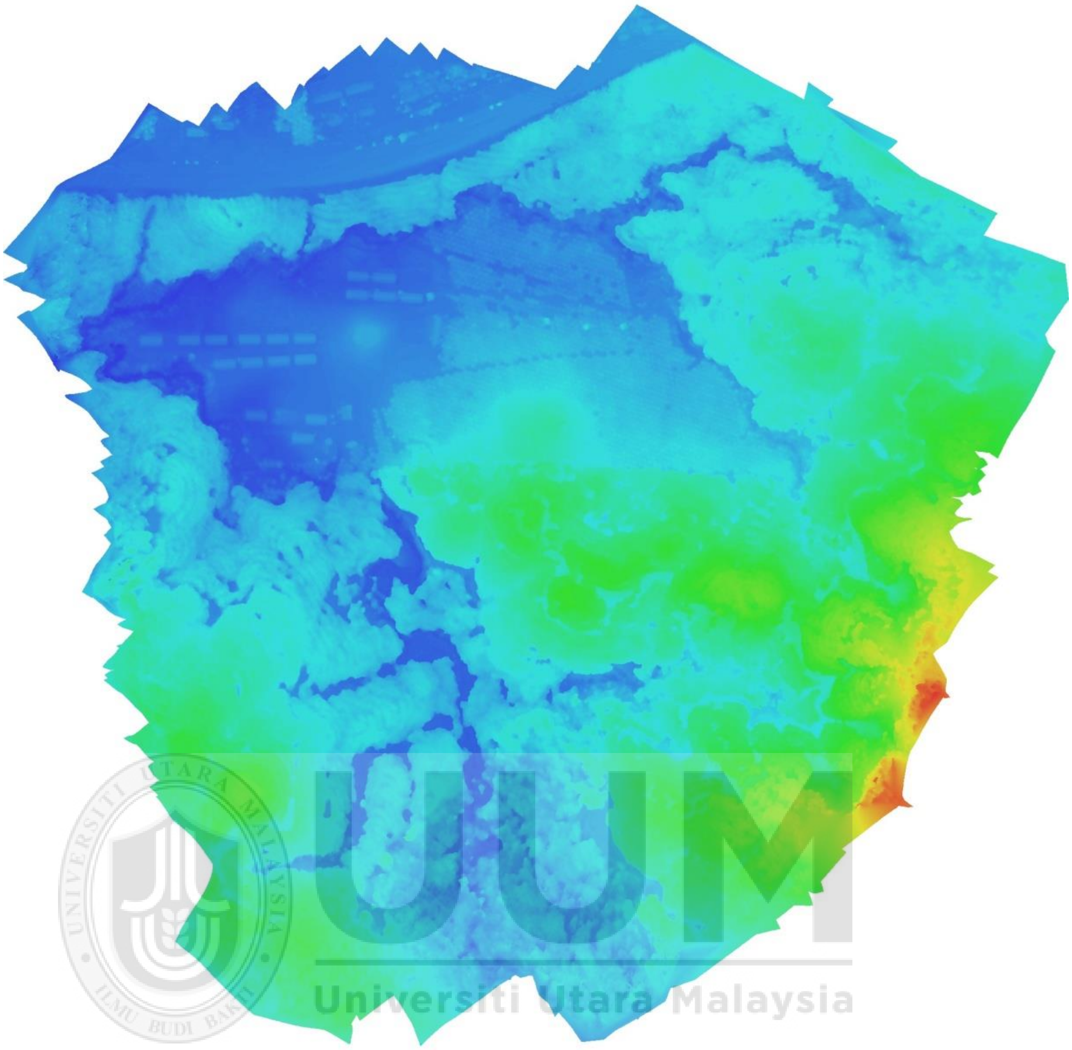

Figure 4.6. DEM data generated from Agisoft photo scan.

Data preparation

The data that were to be produced from Agisoft is raster file in TIFF format that contained information of the terrain. The data will then be clipped to be used in this study. Figure 4.7 shows the menu for clipping the data using ArcGIS software.

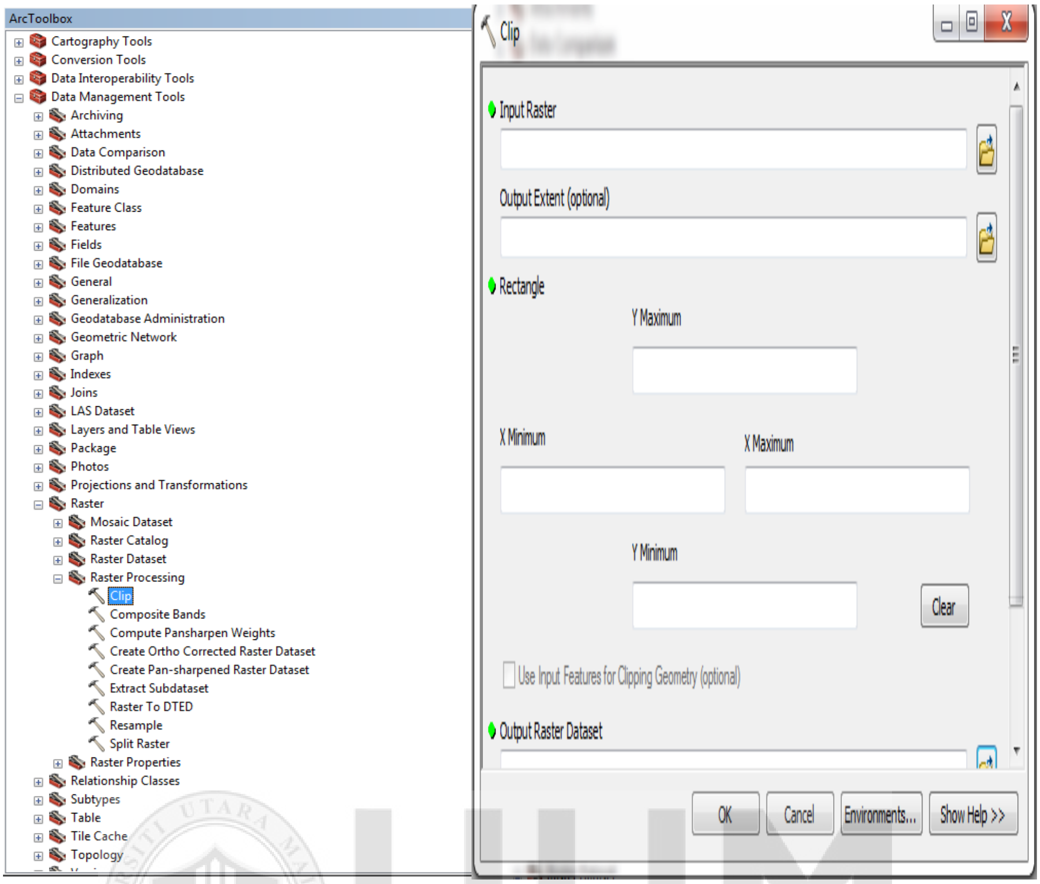

Figure 4.7. Clipping Menu using ArcGIS software.

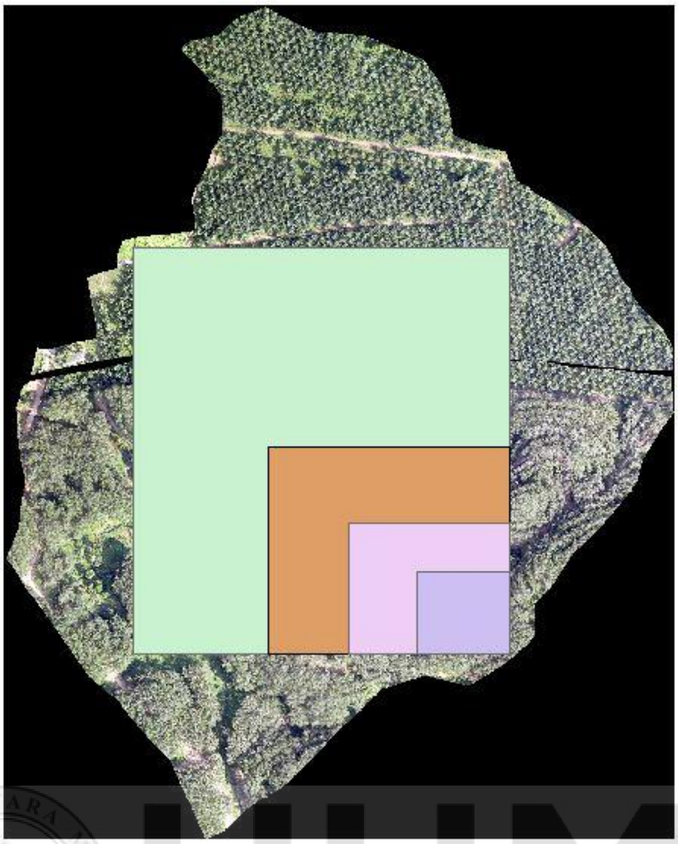

Figure 4.8. The Size of Different Areas Clipped for The Experiments.

Terrain A sizes are 16.927292 hectares and were to include five experiments that are loading time, response time, FPS, CPU usage and memory usage. The clipping of the data also used ArcGIS software.

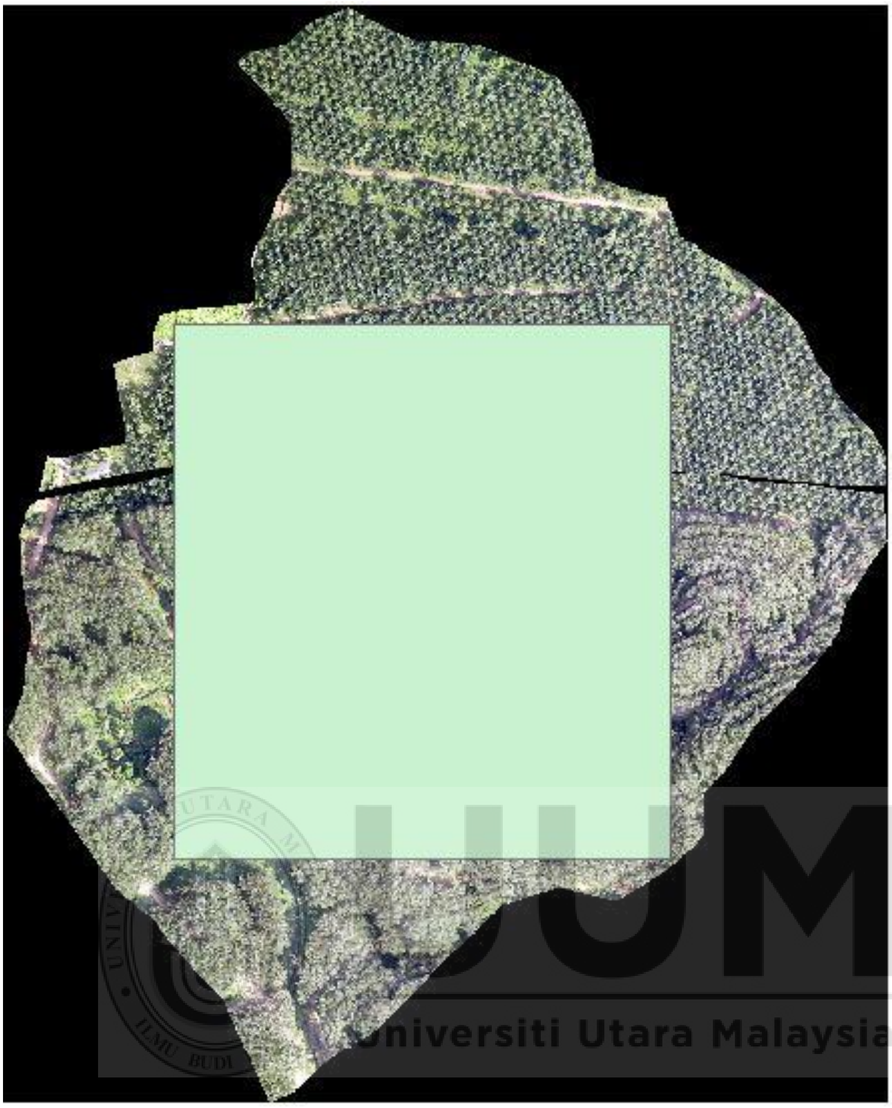

Figure 4.9. Area of Terrain Size A.

The second to fourth data consist of terrain size B is size 5.49895 Hectares, terrain sized C is 2.34673 Hectares and terrain size D 0.841018 Hectares.

Figure 4.10. Area of Terrain Size B, C, and D.

Figure 4.8 to 4.10 show the results of terrain size that is going to be clipped inside ArcGIS, terrain data that have been produced is later loaded into ArcGIS software that converts the terrain data into formats that can be used for this study. The format that is going to be used in this study is named ArcGIS grid float format that can be accessed from ArcGIS toolbox either from ArcScene or ArcMap (refer Figure 4.11).

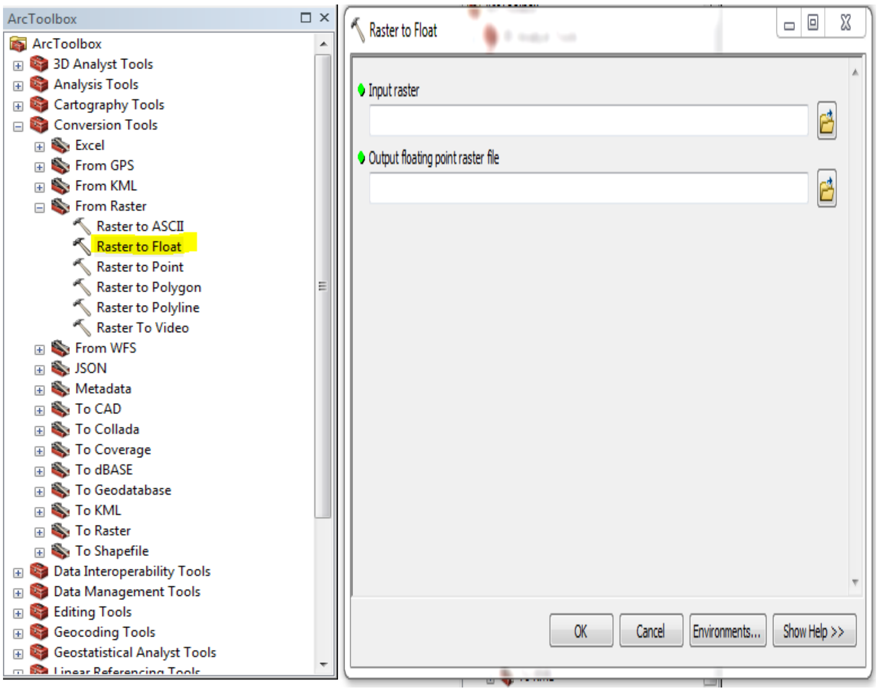

Figure 4.11. ArcGIS Toolbox Converts to Grid float.

4.3 Data generation

After the terrain data have been acquired from ArcGIS. The data is ready to be used in the process of generating 3D terrain visualization with the new enhancing process of algorithm. This new enhancing process of algorithm is not available in other game engine. The next section will discussed details on the process of this algorithm step by step. At the end, the 3D terrain visualization will be generated.

4.3.1 Algorithm of Enhanced 3D Terrain Visualization

HIPO was adopted to explain the breakdown of the process used inside Unity3D to generate terrain visualization. HIPO is divided into 3 categories which are terrain data reading, terrain data conversion, and terrain data processing. The terrain data reading is the process of identifying the input that is needed, terrain data conversion is to convert grid float to Unity3D float and terrain data processing is to applying the

Z data on the terrain. The next sections explain how the enhanced process used the terrain data is delivered, the process that occurs, and what the output should be after the process using HIPO. The first part will look at the process of the script that is used to generate the terrain visualization following the HIPO tools as shown in Figure 4.12 and Figure 4.13, which presents the breakdown of the enhanced process in flow chart formats.

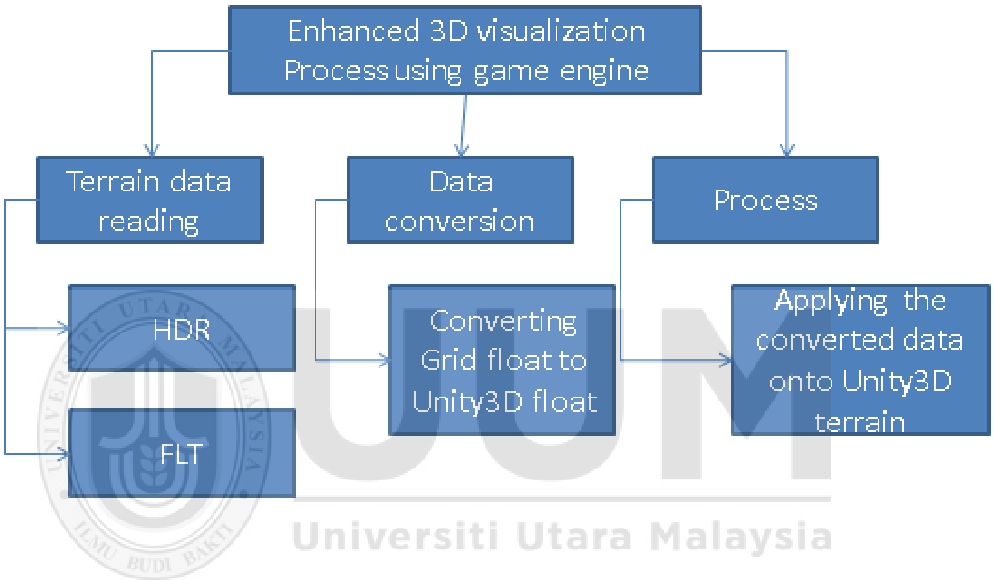

Figure 4.12. Shows the Breakdown of the Enhanced Process in HIPO tools.

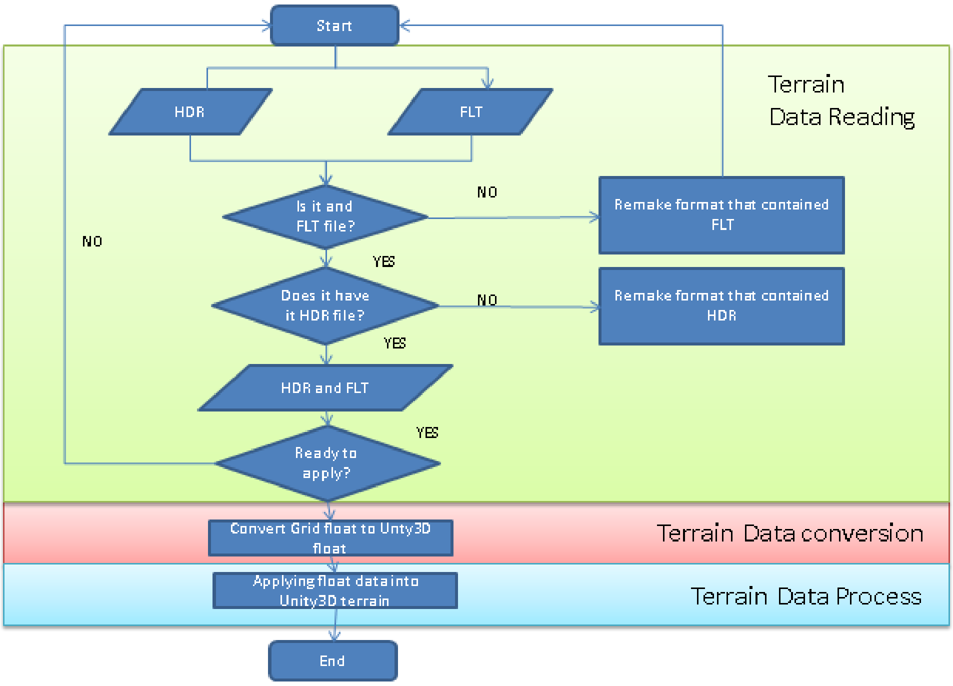

Figure 4.13. Flowchart of the Enhanced Process.

The enhanced process can be divided into terrain data reading, terrain data conversion, and terrain data process. The first is the input; the input for the process is the HDR and FLT data that was generated from ArcGIS. The FLT data contained bytes data in grid float format while the HDR contained the reference on where the bytes are supposed to be inside Unity3D.

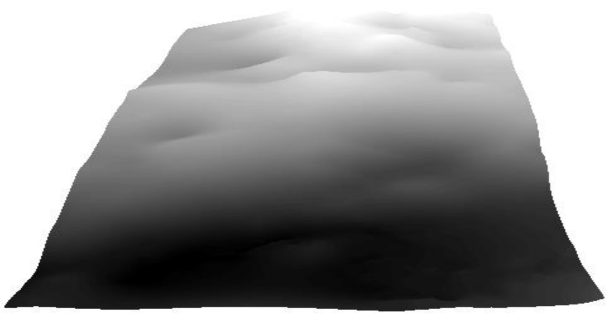

Figure 4.14. Data bytes of FLT opened in ArcGIS.

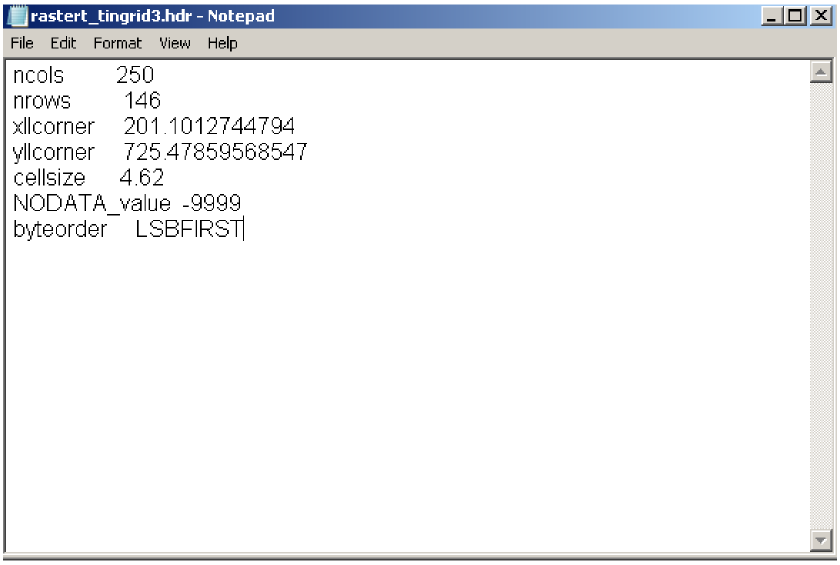

Figure 4.15. Data of HDR opened in notepad.

After the data have been prepared (refer Figure 4.14 and Figure 4.15), the process will continue to use the data as it input process (refer Figure 4.14).

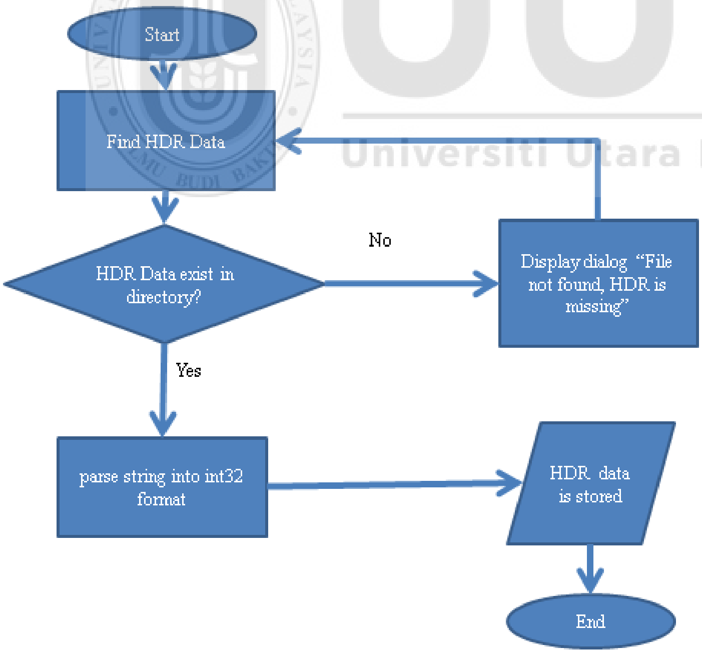

Figure 4.16. Algorithm for getting HDR data.

Input from HDR and FLT will be temporarily stored (refer Figure 4.16 and Figure 4.17).

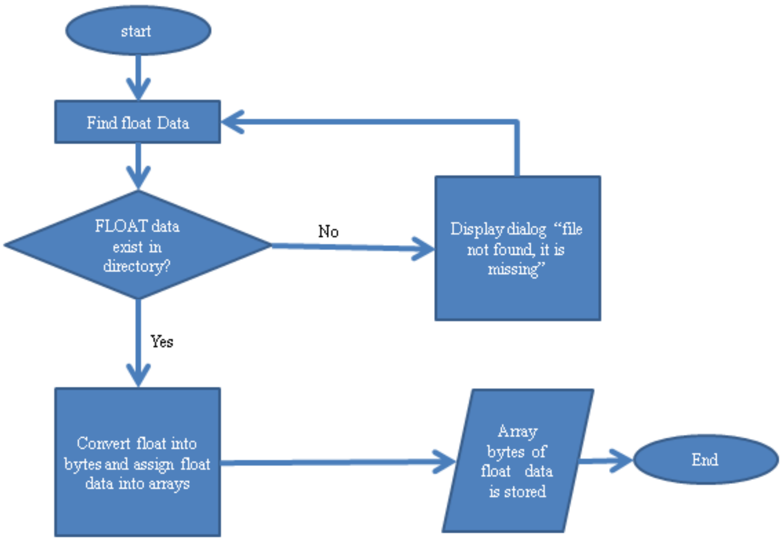

Figure 4.17. Algorithm for getting float data.

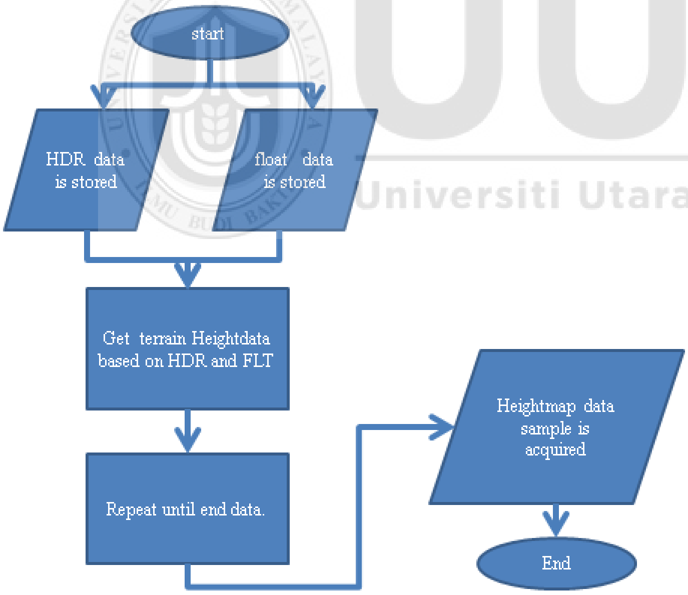

Figure 4.18. Generating sample terrain data.

After both HDR and FLT data have been temporarily stored as shown in Figure 4.18. It will be used for the second part that is the terrain data conversion and generation.

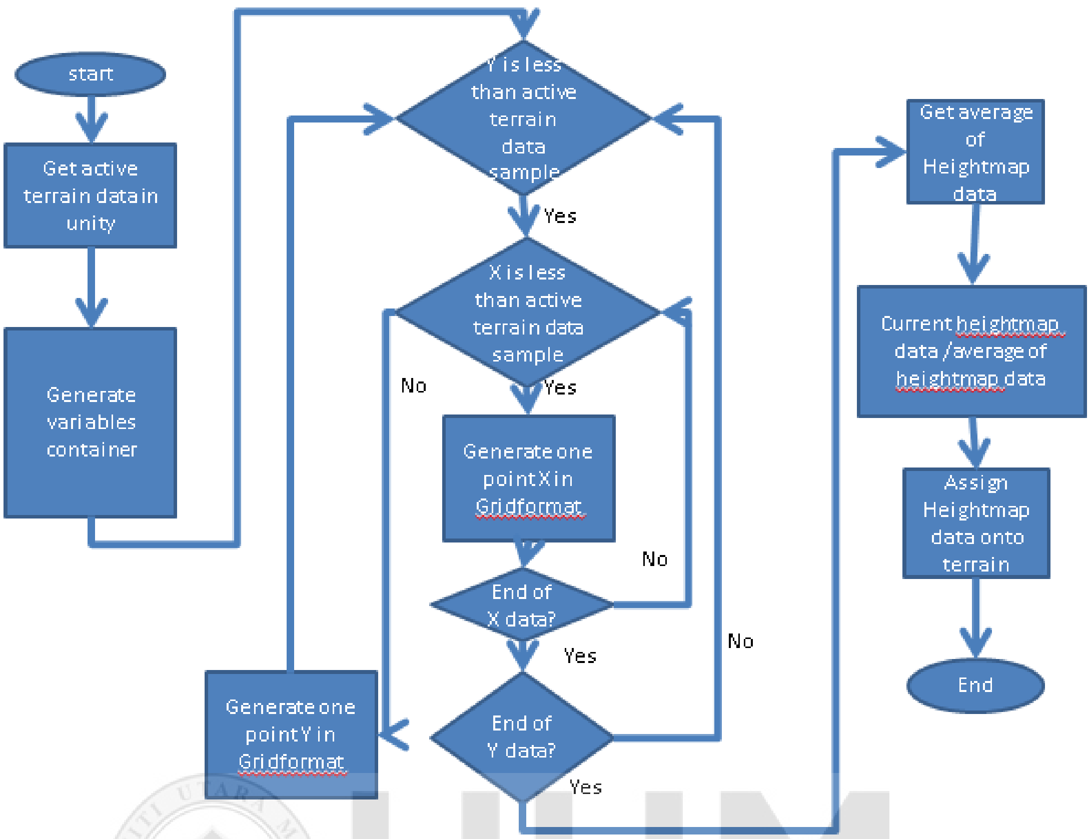

Figure 4.19. Generating terrain data.

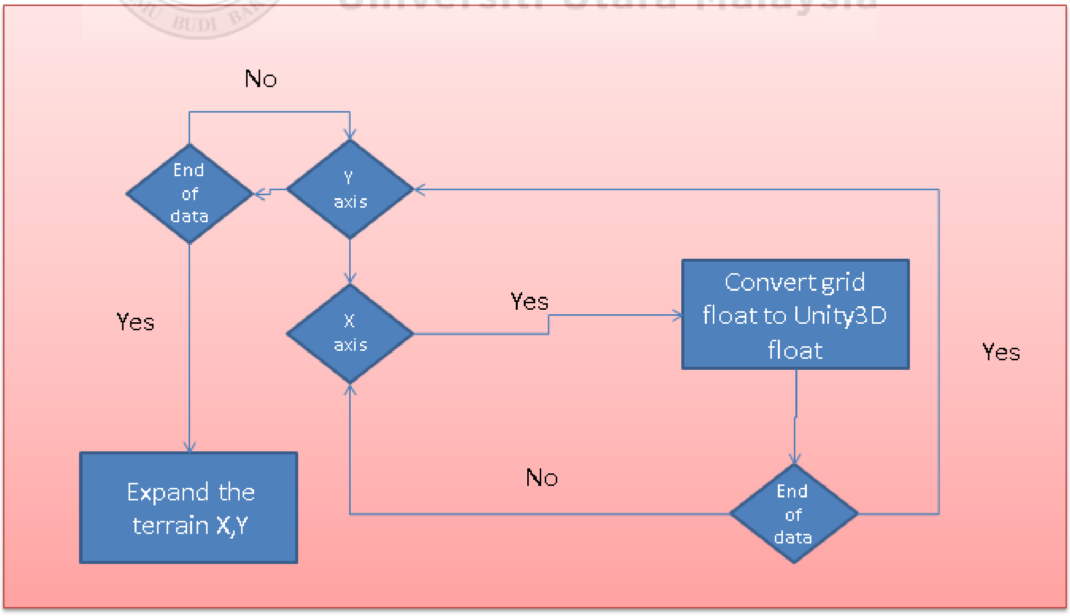

Figure 4.20. The flow of data conversion.

This process of terrain data conversion as shown in Figure 4.19 will load the terrain heightmap sample and converted into grid data for float data in Unity3D. The process will continue as long the data does not exceed heightmap sample data. After the data had finished converted, the average of heightmap data is acquired as shown in Figure 4.20, the average of heightmap data is then divided by the current heightmap sample before it is applied onto the terrain inside Unity3D as shown in Figure 4.21. While Figure 4.22, shows the flow of the process in flow chat format and the output of the process will be applied to the terrain inside Unity3D.

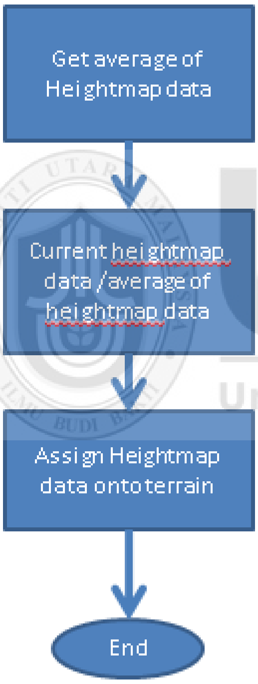

Figure 4.21. The algorithm of the terrain data processing.

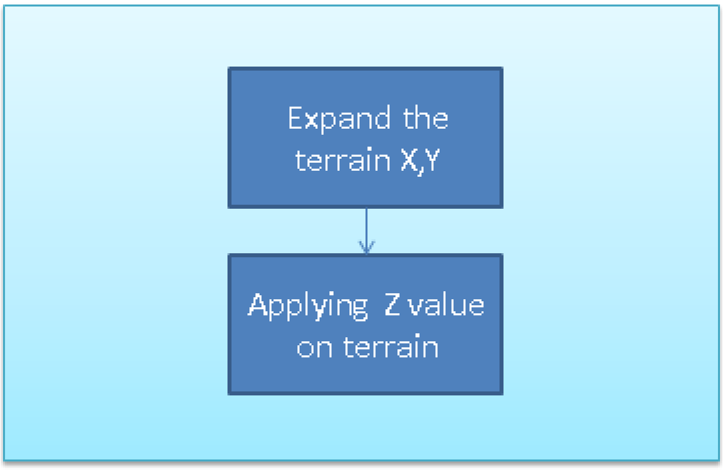

Figure 4.22. The flow of the terrain data processing.

4.3.2 The Process of Image Overlaid and Online Publishing

After the terrain has been loaded, UAV image can be overlaid on the terrain data (refer Figure 4.23). Then the functionality can be added to the project for maneuvering the terrain data in a 3D environment such as camera setting for flying through capability. This function allowed the terrain to be viewed from every angle.

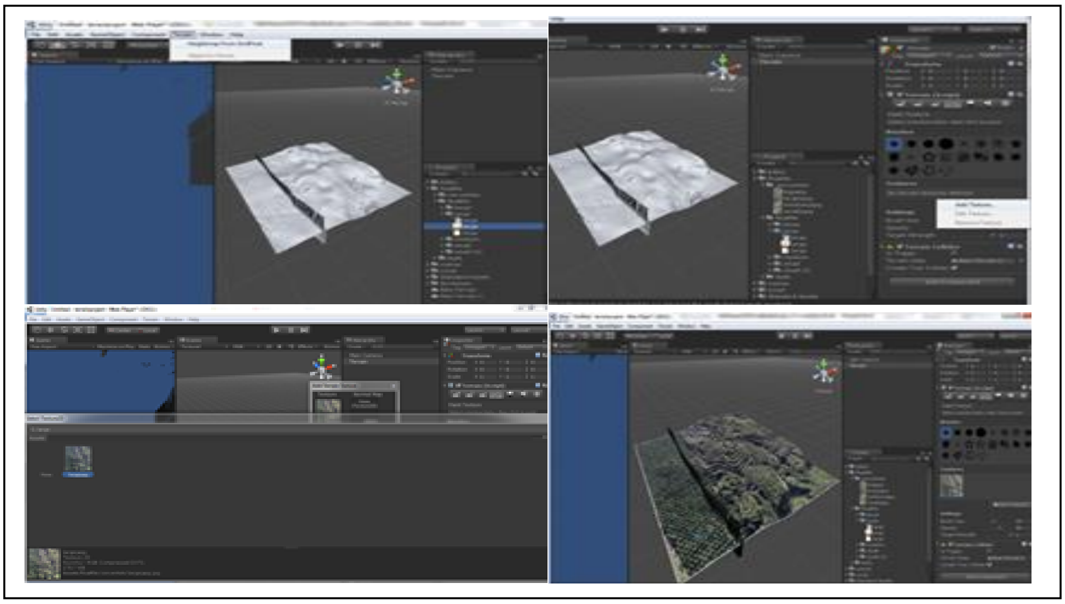

Figure 4.23. The Process of Overlaying the UAV Images into the Terrain Data.

The final process is to publish the system into the web environment. By using the Unity3D game engine this process can be easily performed. Figure 4.24 shows how the system can be published in online environments.

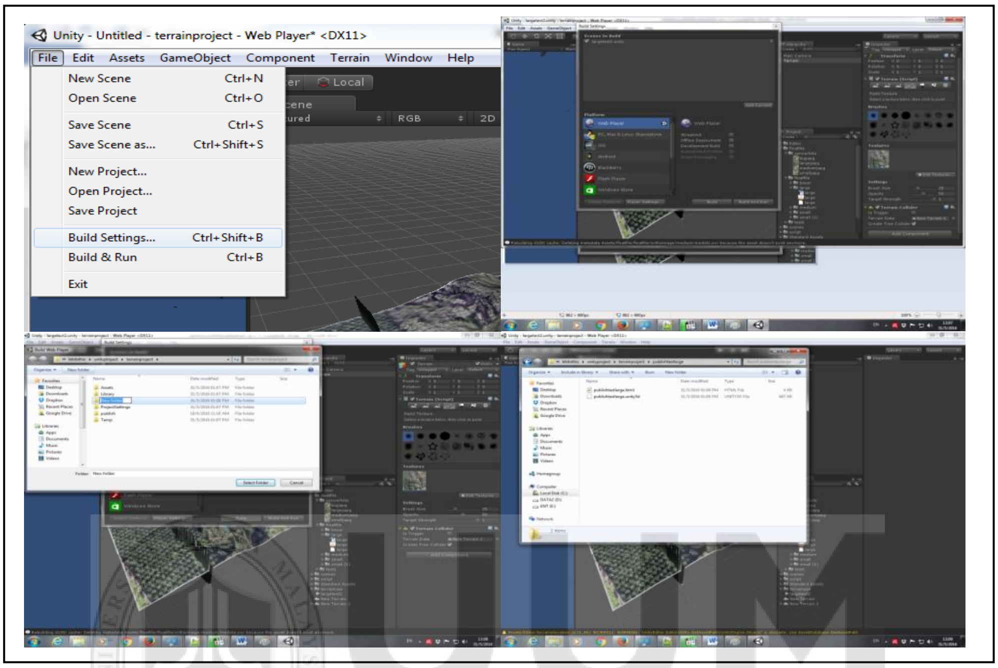

Figure 4.24. The Process Involved In Publishing the Terrain into Online Environment.

4.3.3 Different Size of Terrain Visualization

For the purpose of the evaluation process, the terrain is divided into four different data size which is terrain size A for 16.927292 hectares, terrain size B for 5.49895 hectares, terrain size C for 2.34673 hectares and terrain size D for 0.841018 Hectares. Table 4.1 shows the file size of each terrain in kb before it being published and after published in Unity3D web player format.

Table 4.1

The size of terrain in kb before and after published

Terrain size	Original data size (kb)	Size in Unity3D (kb)
A	14,700	1105,92
B	2,000	1073,15
C	804	954,368
D	572	856,064

In order to compare the performance of enhancing 3D terrain visualization process prototype in Unity3D, the prototype in Unity3D is compared with the two different environments which are the online and offline environment. From the table, the original size of terrain size A is 14,700kb but when it is published in Unity3D it becomes 1105,92 kb. Terrain size B shows an original size of 2,000 kb and 1073,15 kb for Unity3D. While for terrain size C, the original size of terrain is 804kb and 954,368kb for Unity3D. Finally, for terrain size D, the original size of terrain is 572kb and 856,064kb for Unity3D. It can be seen that the file size of each terrain decreased when the size of the terrain in Unity3D also decreased. Figure 4.25 to Figure 4.28 show the results of each terrain data size A, B, C and D when it published in the Unity3D for the online and offline environment.

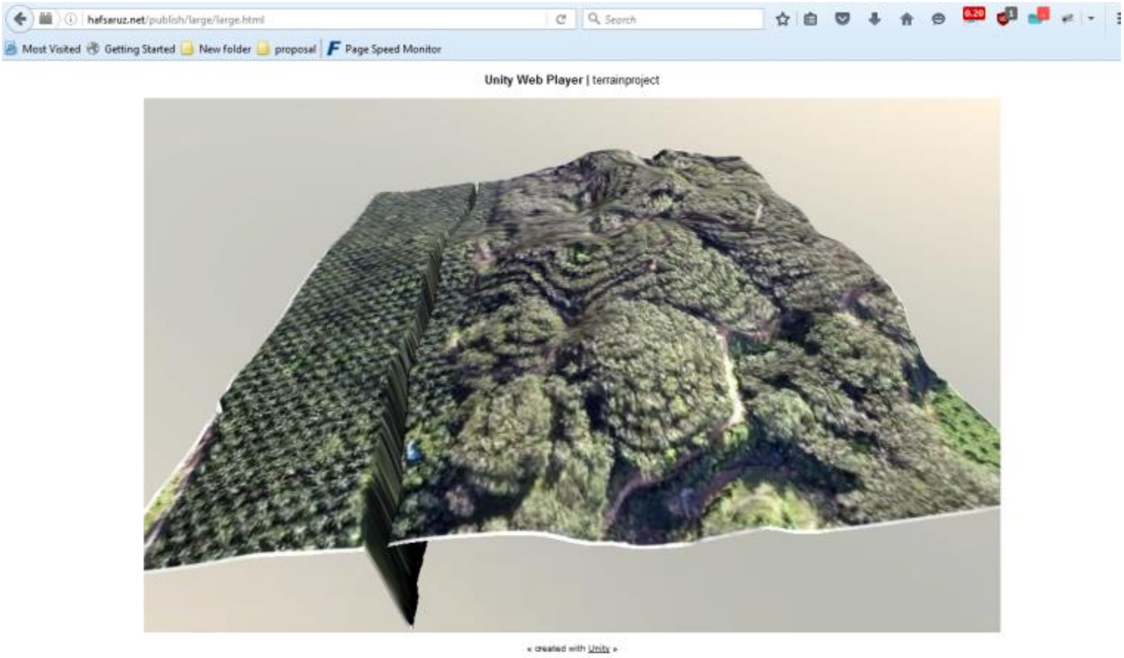

Figure 4.25. Terrain Size A

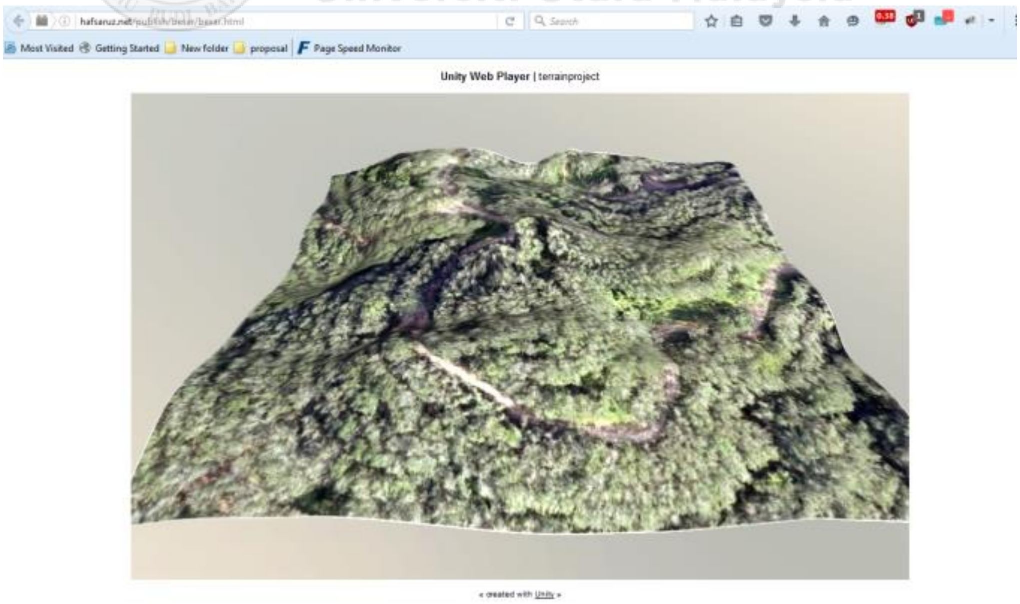

Figure 4.26. Terrain Size B

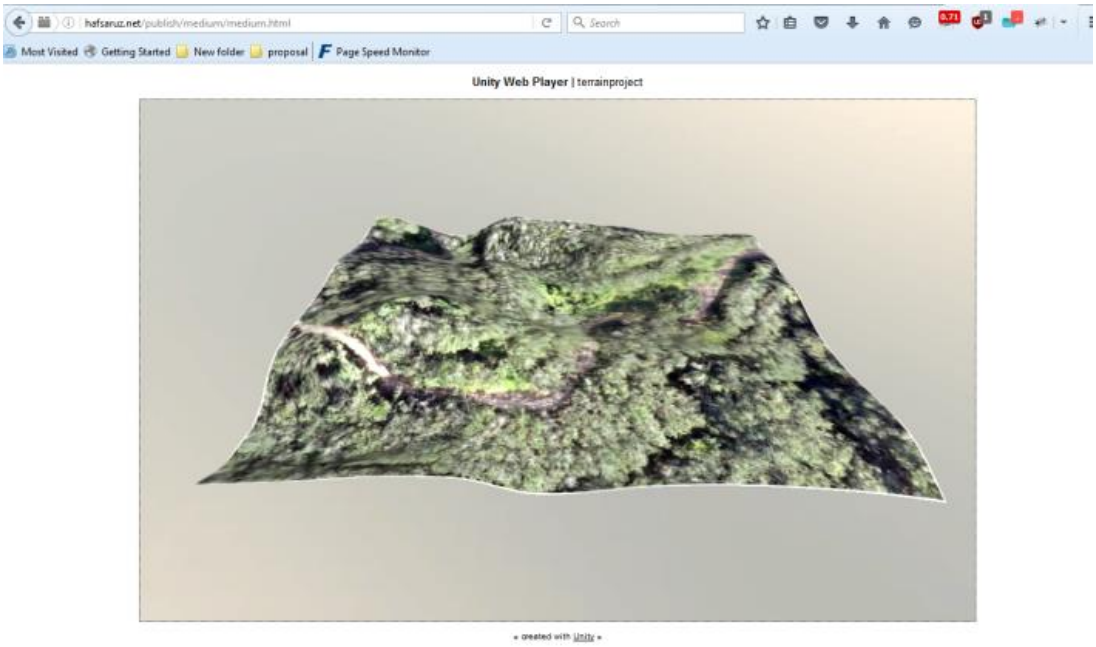

Figure 4.27. Terrain Size C.

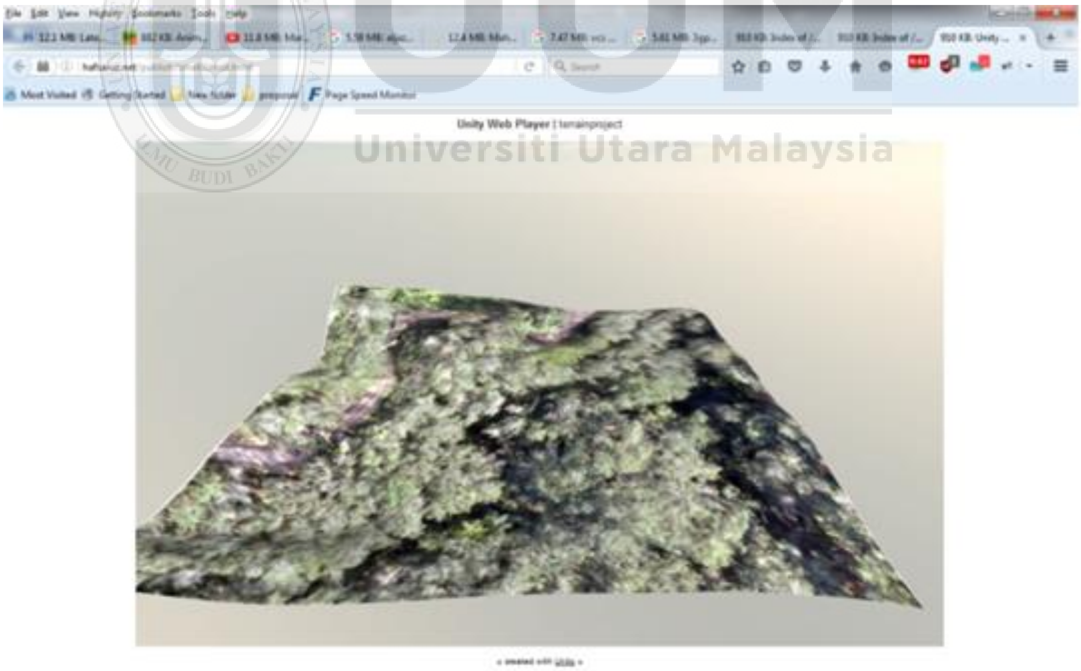

Figure 4.28. Terrain Size D.

4.4 Summary

This chapter discussed the development of web-based 3D terrain visualization by using game engine. The development process is divided into three phases which are data acquisition, data preparation, and data generation. The acquisition is the data collected for this study using UAV data that was collected. The second is the preparation in this data preparation can be seen as raw terrain data processing and unity 3D terrain data processing and finally the data generating where the process loaded the terrain data inside Unity3D, display the terrain visualization inside Unity3D, as well as applying the UAV images onto the terrain inside Unity3D before published in web environment. This chapter contributes to the main finding of this study whereby the new algorithm for enhancing 3D terrain visualization process using game engine has been introduced. The algorithm consists of three processes which are terrain data reading, terrain data conversion and terrain data process which explained details in this chapter.

CHAPTER FIVE
RESULTS AND DISCUSSION

5.1 Introduction

This chapter discusses the results of this research. The results are divided into two parts; the first part discusses the evaluation of the performance of enhancing 3D terrain visualization process prototype in Unity3D in online and offline environments, and the second part discusses the experiments, the results of which will serve as a guide to developers of visualization. The detailed discussions of all these two parts are presented in the following sections.

5.2 The Evaluation of Enhanced 3D Terrain Visualization Process Prototype

The performance of enhancing 3D terrain visualization process prototype in Unity3D in online and offline environments was compared based on loading time, response time, FPS, CPU usage, memory usage and comparison of all measurement for each terrain data size in Unity3D in an online and offline web environment.

5.2.1 Comparison of the Loading Time

The first comparison is made based on loading time. The measurement of loading time is performed by using "Page speed monitor”. The measurement was performed by collecting data four times and the average value of the four tests was calculated. Figure 5.1 shows the loading time record using "Page speed monitor". Figures 5.2 shows the result collected from the test.

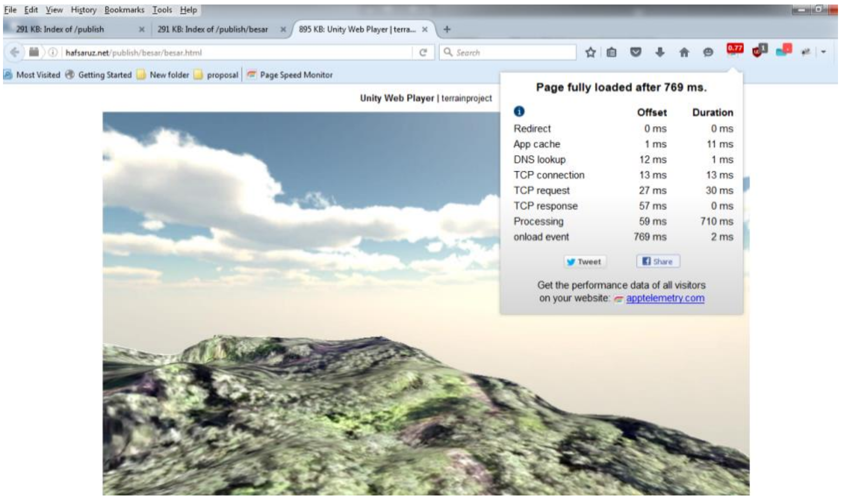

Figure 5.1. The Loading Time Recorded using Page Speed Monitor.

The result of the comparison is divided into two sections, the first section is the results in the form of a table (Table 5.1) and the second section is the results in the form of a graph (Figure 5.2).

Table 5.1

Comparison of loading time for online and offline environment

Terrain Size	Loading time (ms)	
	Online	Offline
Terrain size A	0.136	0.01
Terrain size B	0.769	0.01
Terrain size C	0.719	0.01
Terrain size D	0.657	0.01

Table 5.1 shows the results of loading time for enhancing 3D terrain visualization process prototype in online and offline environments. Since both of prototypes are measured in two different environment that is online and offline, the comparison can be seen clearly. Results from the prototype revealed that the value can change depending on the status of the browser and the user's computer. The results for online terrain size A stated the lowest value while the terrain size B stated the highest value. The results for each terrain size are changing based on the terrain data size. This is due to the reflection in the internet service and the current user activity

in the browser as well hidden service that runs during the session. While the results from offline environment show a consistent loading value for each terrain data size (refer Figure 5.2) this is because there is no online transfer happening for page speed monitor to detect. This can be concluded that the terrain data size does not affect the loading time of enhancing 3D terrain visualization process prototype in the Unity3D game engine. however, testing for Unity3D web player works both online and offline environments revealed that online environment Unity3D required authentication from the Unity3D server before it can load. This is different from offline there are no needed for Unity3D web player to be made, other than that, web player file also requires decompression. All of this can affect the value of loading time for visualization in an online environment. Figure 5.2 shows the comparison results of loading time in terms of a chart.

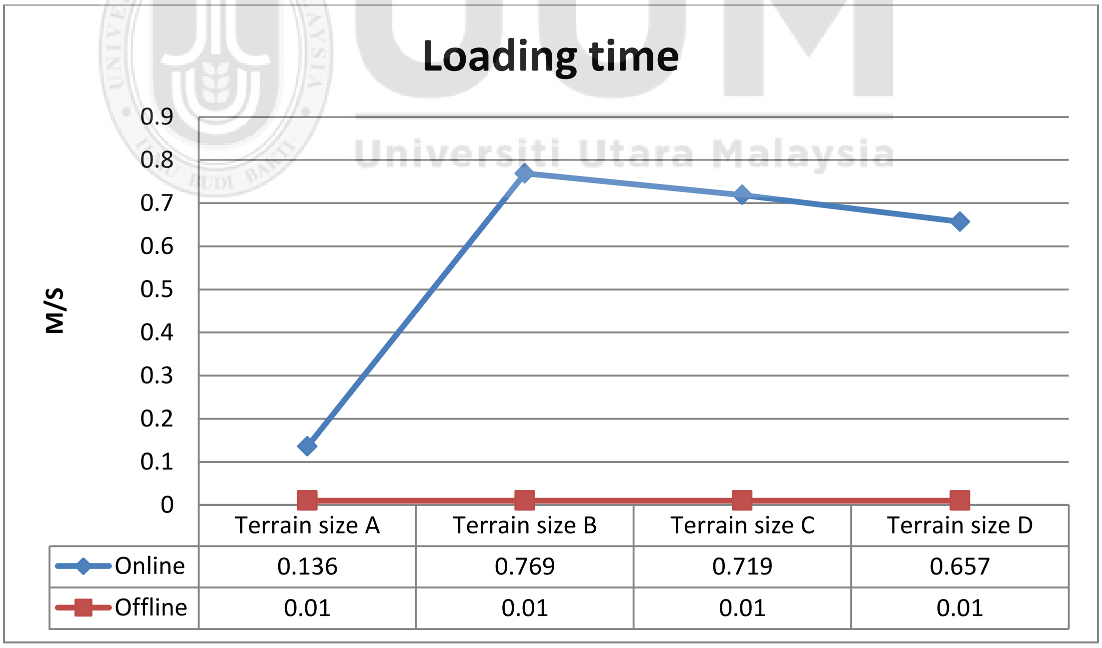

Figure 5.2. The Loading Time Graph for Comparison of the Online and Offline Environment.

From Figure 5.2, we can easily recognize that from the whole value of loading time recorded, the performance of enhancing 3D terrain visualization process prototype in offline environments in terms of loading time is better as compared to online environments. However, this value still falls under 18 seconds which is acceptable for online environments to be loaded (SEOchat, 2017).

5.2.2 Comparison of the Response Time

The second comparison is made based on response time. The measurement of response time is also performed using similar "Page speed monitor" as mentioned earlier in Chapter Three. The result of the comparison is divided into two sections, the first section is the results in the form of a table (Table 5.2) and the second section is the results in the form of a graph (Figure 5.3).

Table 5.2

Comparison of response time for online and offline web environment

Terrain Size	Response time (ms)	
	Online	Offline
Terrain size A	11	0.01
Terrain size B	57	0.01
Terrain size C	72	0.01
Terrain size D	72	0.01

Table 5.2 shows the results of response time for enhancing 3D terrain visualization process prototype in the online and offline environment. The results of response time gathered from this measurement shows that online environment data recorded a longer response time value compared to offline. Figure 5.3 shows that the terrain size-A recorded the lowest response time value which is 11 ms among the other terrain size. However, for offline, show a consistent value of 0 ms response time.

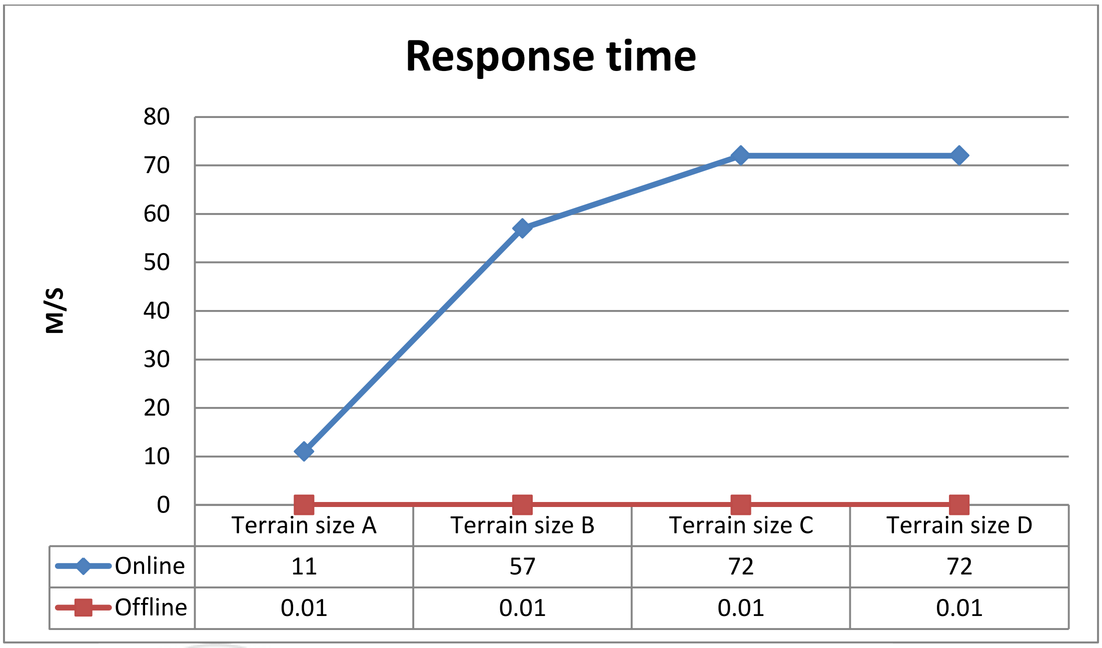

Figure 5.3. The Response Time Graph for Comparison of the Online and Offline Web Environment.

Figure 5.3 shows the results of each of the terrain size clearly. These results can be understood that both online and offline have a different response time value. Most of the response time value for offline environment revealed the lowest compared to online environment. This situation is the case may be due to the online environment requiring authentication for visualizing it on the user browser. This can affect the value of response time for the online environment. While in offline environments authentication visualization is not required. In a study by Hussain, Wang, Toure, and Diop (2013) mentioned that average response time of an online website is about 3000 MS and the result of this is averaging 80 MS which is much better than expected.

5.2.3 Comparison of the Frame per Second (FPS)

The third comparison is made based on FPS value. The measurement of FPS for both offline and online is performed using Firefox resource monitor. The motions appear smoother when the FPS value is higher. That is why when the value of fps is much higher; the visualization of the system is smoother. Figures 5.2 shows the result collected from the *F*irefox performance test.

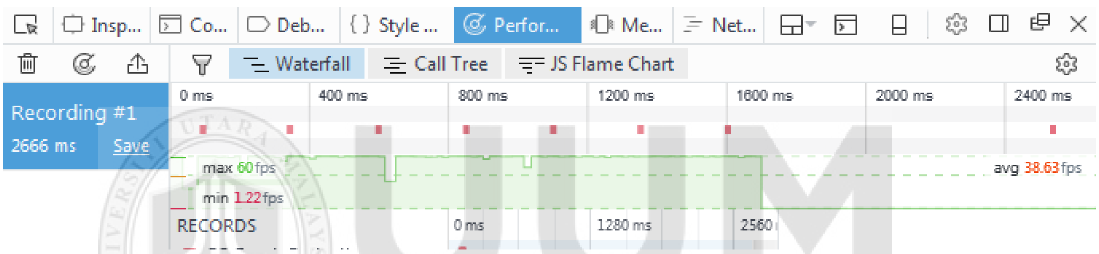

Figure 5.4. The FPS Value Recorded Using Firefox Performance Test.

The result of the comparison is divided into two sections, the first section is the results in the form of a table (Table 5.3) and the second section is the results in the form of a graph (Figure 5.5).

Table 5.3

Comparison of FPS value for the online and offline web environment.

Terrain Size	Frame per second value	
	Online	Offline
Terrain size A	57.03	53.68
Terrain size B	51.77	55.52
Terrain size C	50.95	57.78
Terrain size D	52.44	54.84

Table 5.3 shows the results of FPS value for enhancing of 3D terrain visualization process prototype in online and offline. The result shows that the FPS value for all of the terrain size is high for offline environments except for terrain size-A which revealed the lowest as compared to online. The average FPS value compares to online for all-terrain size in online is about 53.05 while the average FPS value for offline is 55.45. It can be concluded that the FPS value for offline is better compared to online environments.

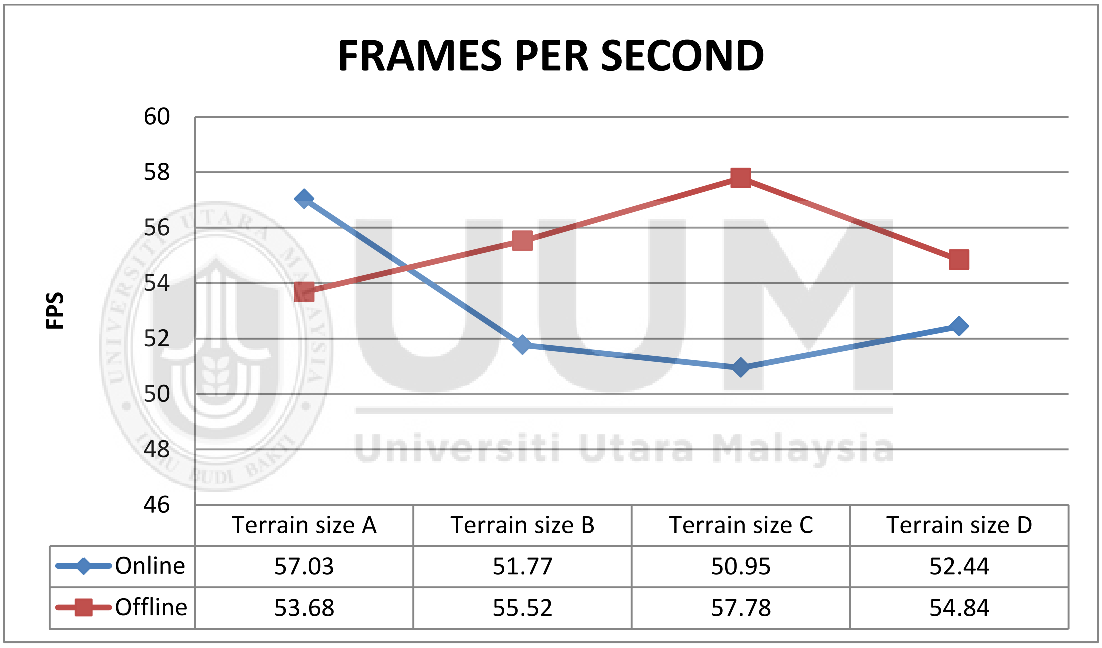

Figure 5.5. The FPS graph for Comparison of Online and Offline.

Figure 5.5 shows the results of each of the terrain size clearly. It shows that FPS value for online environment is not stable which showing the results inconsistently from terrain size A to terrain size D. This may be due to compression and decompression of its data after publishing in web player which affecting the FPS value while in online environments. In a study by Janzen and Teather (2014), it was mentioned that average FPS value for any game is around 45 FPS while this study has achieved a result more than average in both online and offline environments.

5.2.4 Comparison of the CPU Usage

The fourth comparison is made based on CPU usage. The measurement of CPU usage is performed using "Process Explorer". This software measured and viewed the value of CPU usage for each terrain data. Figure 5.6 shows the value of CPU usage in term of percentage collected using "Process Explorer".

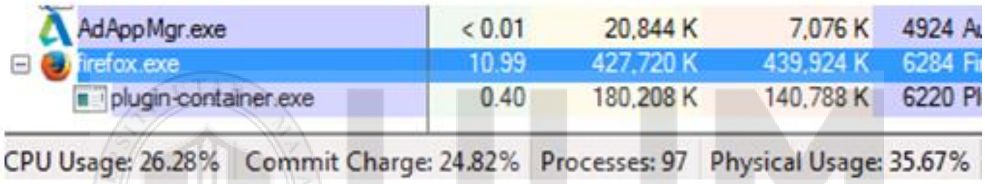

Figure 5.6. The CPU Usage Value Recorded Using Process Explorer

The result of the comparison is divided into two sections, the first section is the results in the form of a table (Table 5.4) and the second section is the results in the form of a graph (Figure 5.7).

Table 5.4

Comparison of CPU usage for the online and offline environment.

Terrain Size	CPU usage value (%)	
	Online	Offline
Terrain size A	10.99	7.86
Terrain size B	12.03	6.5
Terrain size C	11.73	5.6
Terrain size D	11.36	4.8

Table 5.4 shows the results of CPU usage for enhancing 3D terrain visualization process prototype in online and offline environments. The CPU usage is recorded

based on the highest percentage value while running the system. The results of CPU usage gathered from this measurement shows that online data environment recorded the highest percentage value compared to offline. Figure 5.3 shows that the terrain size B recorded the highest percentage CPU usage which is 12.03% among the other terrain size in an online environment. However, for offline, terrain size D recorded the lowest percentage CPU usage which is 4.8% compared to the other terrain size.

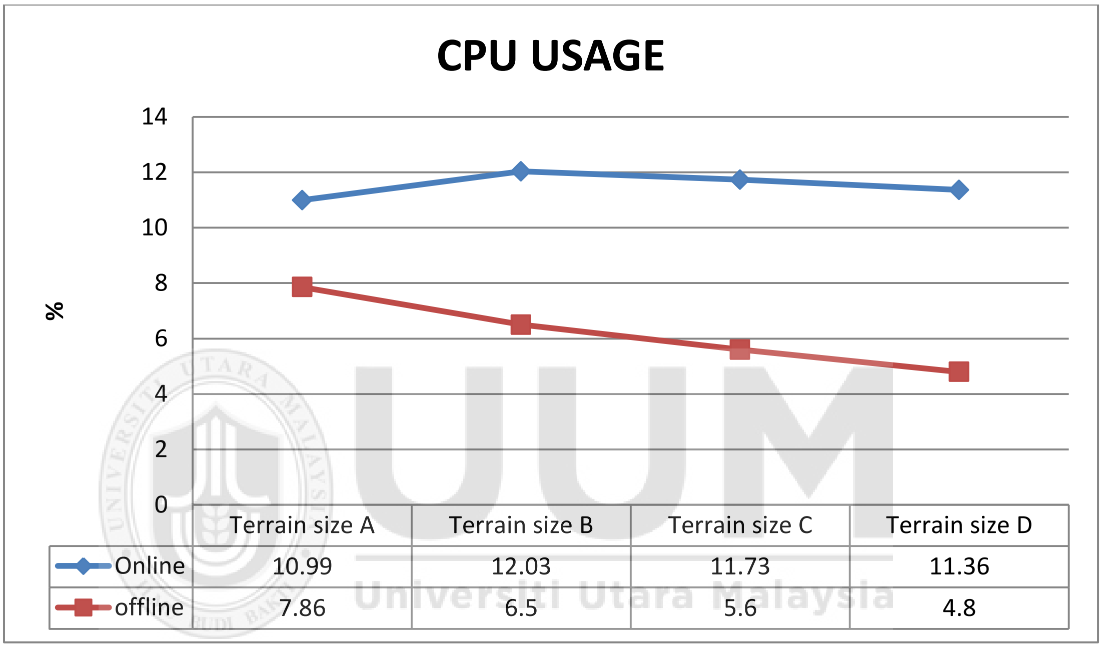

Figure 5.7. The CPU Usage Graph For Comparison Of The Online And Offline Web Environment.

Figure 5.7 shows the results of each of the terrain size clearly. It shows that the percentage of CPU usage for online as well as offline is not stable which presents the results inconsistently from terrain size A to terrain size D. This may be due to compression and decompression of its data after publishing in the web player. That affects the CPU usage percentage. While in online environments, features required like window network service to be activated compared to offline environment. Kumar et al. (2008) uses havoc physic as engine for visualizing environment of Second life online game and found that the average CPU usage is 27% besides that, Juarez, Schonenberg, & Bartneck (2010) uses unreal engine 2 to check CPU performance of its prototype and found that the average of his CPU usage is 0.5%. This can be concluded that the prototype has acceptable CPU usages value while in online and offline environments.

5.2.5 Comparison of the Memory Usage

The fifth comparison is made based on memory usage. The measurement of memory usage is also performed using "Process Explorer". The value of memory usage is recorded in kilobytes as the "Process Explorer" shows the results in kilobytes. The measurement is recorded based on the single tab that loaded each of the terrain data from the server into Firefox browser.

The result of the comparison is divided into two sections, the first section is the results in the form of a table (Table 5.5) and the second section is the results in the form of a graph (Figure 5.8).

Table 5.5

Comparison of memory usage for the online and offline environment.

Terrain Size	Memory usage (Kb)	
	Online	Offline
Terrain size A	180,208	139,704
Terrain size B	196,572	138,584
Terrain size C	204,028	134,496
Terrain size D	212,068	134,936

Table 5.5 shows the results of memory usage for enhancing 3D terrain visualization process prototype in the online and offline environment. The results of memory usage gathered from this measurement shows that online environment data recorded the highest value of memory usage compared to offline environment. Figure 5.8

shows that the terrain size D recorded the highest value of memory usage which is 212,068Kb among the other terrain size in Unity3D. However, for the offline environment, terrain size C recorded the lowest value of memory usage which is 134,436Kb compared to the other terrain size. However, this result does not represent the total data needed to be cache inside the computer. The value of memory usage from this results also represent as RAM usage for each of the terrain data.

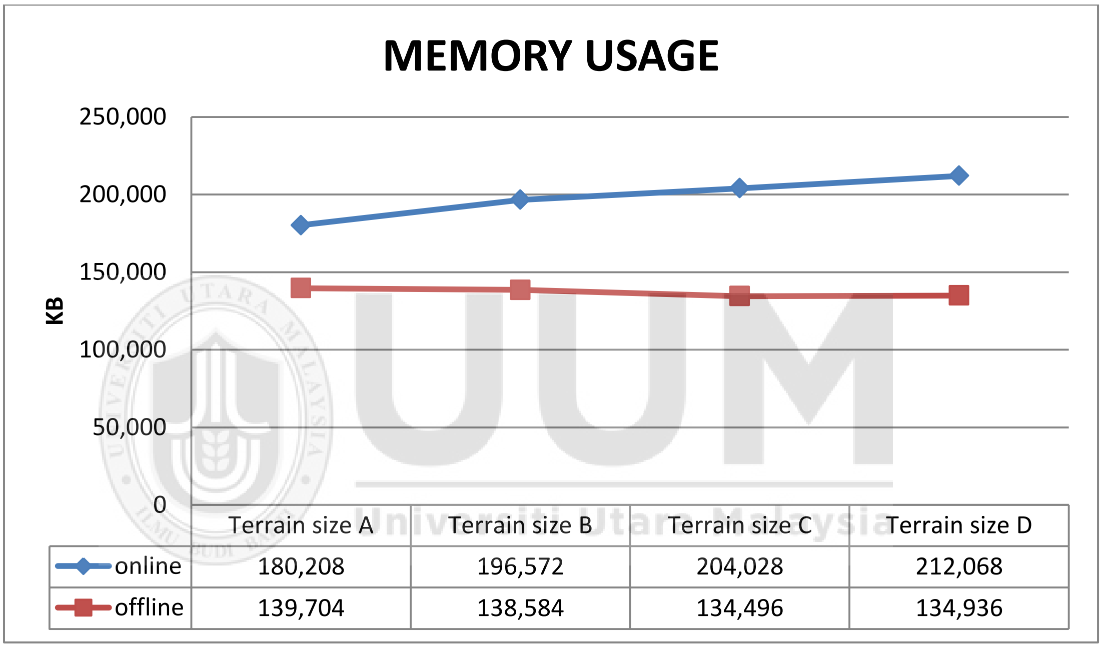

Figure 5.8. The Memory Usage Graph for Comparison of Online and Offline Environments

Figure 5.8 shows the difference between each of the memory usage clearly. It shows that the value of memory usage for online environment keeps increasing from terrain size A to terrain size D. Besides that, the value of memory usage for both online and offline on all different terrains does not follow the “biggest data should have more memory usage” rule when published in online environments. Indraprastha and Shinozaki (2009) in their research mentioned that higher pixel images mean higher memory usage. It was highlighted that 1000 pixel image would use average around

450 000 KB of memory. Compared to this prototype revealed that memory usage is still lower and acceptable for each of the terrain sizes that is used.

It can be concluded that most of the results for each of the measurement loading time, response time, FPS, CPU usage and memory usage for offline are better compared to results for the online environment. It means that the performance of in offline environment is better compared to online environments. This situation happens maybe due to Unity3D's need for maximum LZMA compression when publishing the files for web users, as well the resources needed to maintain to access the data in online environments. However, offline does not require for additional resources used for publishing its files to the online environment. Figure 5.9 shows the process of LZMA compression performed in Unity3D.

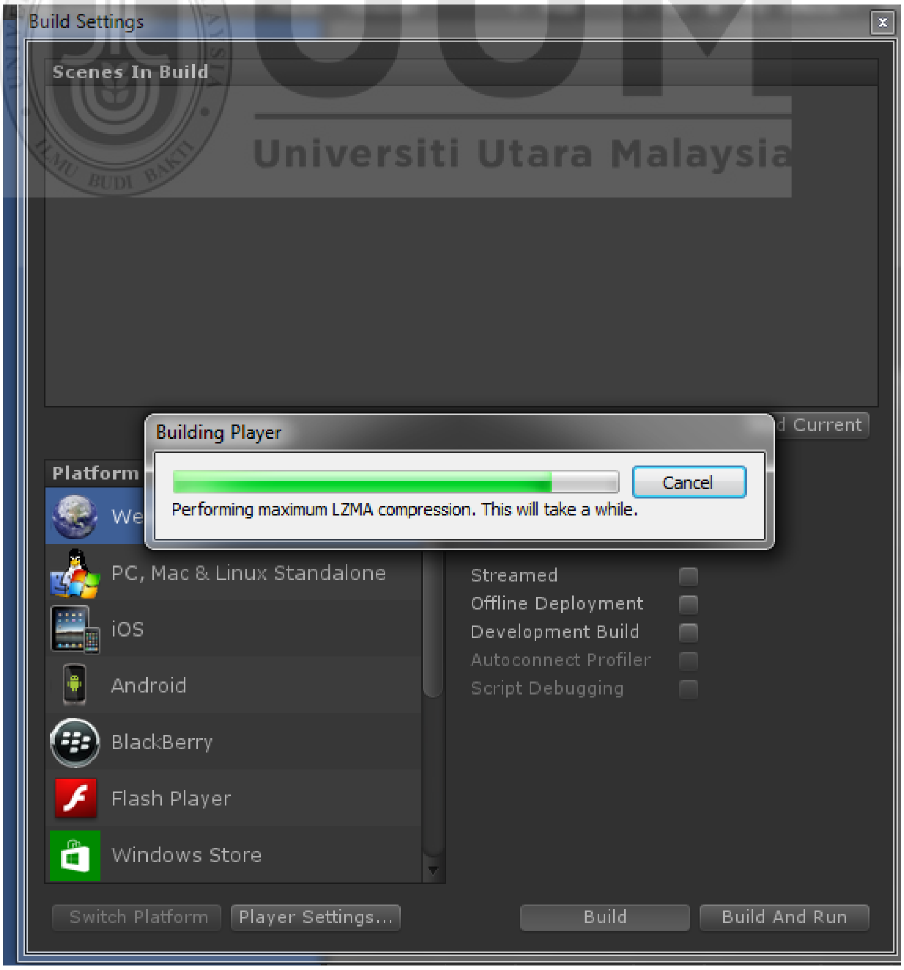

Figure 5.9. The Process of Compression Using LZMA in Unity3D

The LZMA compression process is required to use more memory (Herman, 2016) while processing the files for web publishing. Besides that, Collin (2005) also mentioned that the process of compression required more memory and CPU usage. The resources needed to retrieve data in the online environment. This is different from offline which do not require additional resources to load the online environment. Both offline and online environment also include dependencies such as supported extension file that is compatible. That is why the performance of offline is better compared to the online environment.

5.2.6 Comparison of All Measurement for Each Terrain Data Size In Unity3d

This comparison collected all of the measurement results being tested before like loading time, response time, FPS, memory usage and CPU usage in Unity3D and viewed in the single table. But this time, the results for loading time is furthered analyze and tested at two different times which are during office hours (in) and after office hours (out). Table 5.6 shows the results of this comparison.

Table 5.6

The results for comparison of all measurement for each terrain data size in Unity3D

Criteria	**Hafsaruz.net**							
	A		**B**		**C**		**D**	
Data size	1,105,920kb		1,073,152kb		954,368kb		856,064kb	
Loading time office hour (Off) and after office hour (after) (ms)	in	after	in	after	in	after	In	after
	136	477	769	1441	719	1010	657	1073
Response time (ms)	in	out	in	out	in	out	In	out
	11	12	57	59	72	61	72	208
FPS value	58.11		57.54		56.45		52.44	
CPU Usage(%)	10.99		12.03		11.73		11.36	
Memory usage(kb)	180,208		196,572		204,028		212,068	

The results show that the data size for terrain A is larger compared to another terrain size. While terrain size D recorded smaller data compared to the other terrain. Besides that, the loading time for terrain size A during office hours was 136ms while after office hours were 477ms. The loading time during office hours for terrain size B was 769ms and after office hours was 1441ms. For terrain size C, the loading time during office hours was 719ms while after office hours was 1010 ms. For terrain size D, the loading time during office hours was 657ms while after office hours was 1073 ms. Comparing all of the terrain sizes, terrain size A has the fastest loading time during office hours and after office hours compare to the other terrain size. While terrain size B has the slowest loading time during office hours and after office hours compare to the other terrain size.

After further investigation, the result of terrain A is faster compared to other terrain sizes. This is because terrain A contained noise during the process of loading inside Unity3D. The script that was used to interpret the data and additional noise inside the data are also visualized inside Unity3D. For all of the terrain size, the loading time after of office hours is faster compared to loading time during office hours. This is true because the traffic for internet bandwidth is heavy during office hours compared to out of office hours. More people or users tend to access the internet during office hours compared to after office hours. This is because Unity3D web player sends a request to unity server first to enable a view of web player contents which is the core of this terrain visualization. The results of response time for terrain size A during office hours was 11ms while after office hours was 12ms. While the response time office hours for terrain size B was 57ms and after office hours was 59ms. For terrain size C, the response time during office hours was 72ms while after office hours was 61ms. For terrain size D, the response time during off office hours was 72ms while after office hours was 208ms. Comparing all of the terrain sizes, terrain size A has the fastest response time during office hours and after office hours compare to the other terrain size. While terrain size D has the slowest loading time during office hours and after office hours compare to the other terrain size.

This could be the same as response times of the terrain data. However, both loading time and response time loaded things by below 1500 ms which is good based on the article written by Shaun Anderson (2015), which explains how web optimization works. In the article, the author explained that a webpage should load on average of four seconds. And the result shows it is 4000 ms.

In terms of frames per second; terrain size A had the best FPS value which is 58.11 compared to terrain size B, C, and D which had 57.54, 56.45 and 52.44 respectively. This condition should be the false cause in a normal situation where the data size is bigger, the value of FPS should be slower. The FPS is basically linked to user PC performance and for this test, the PC that was used is good considering all of the results of the terrain gathered an average of 50 FPS value.

In terms of CPU usage; terrain size A had the best percentage of CPU usage which is 10.99% compared to terrain size B, C, and D which had 12.03%, 11.73%, and 11.36% respectively. As for CPU usage, the test was conducted on Firefox using process explorer the data is captured on the highest peak inside process explorer. The results gathered that Firefox uses an average of 10 % of a computer CPU for viewing the terrain data. This affects the percentage of CPU usage.

Lastly, in terms of memory usage; terrain size A had the best of memory usage which is 180,208Kb compared to terrain size B, C and D which had 196,572Kb, 204,028Kb and 212,068Kb respectively. It can conclude that, when the size of a terrain decreases, the value of memory usage increases. The result shows that average memory usage of all of the terrain size is around 200Kb which is equal to 200Mb of memory usage from 8 Gb of RAM. Firefox browser itself consumes around 1.5 Gb of memory.

5.3 Experiment Conducted

This section discusses experiment based on contour interval. The result of this study is from three unity web player format that was created from different contour interval data which are 5m, 3m, and 1m. The output was measured based on data size, loading time (in-office and after hours), response time, visualization quality, and fps. The first output is the 5m interval. Figure 5.11 shows the image of this output. The results of this image reveal an "Acceptable" visualization quality where the area inside the circle can be seen clearly but difficult to identify the slopes. Overall the area of oil palm plantation can be viewed clearly.

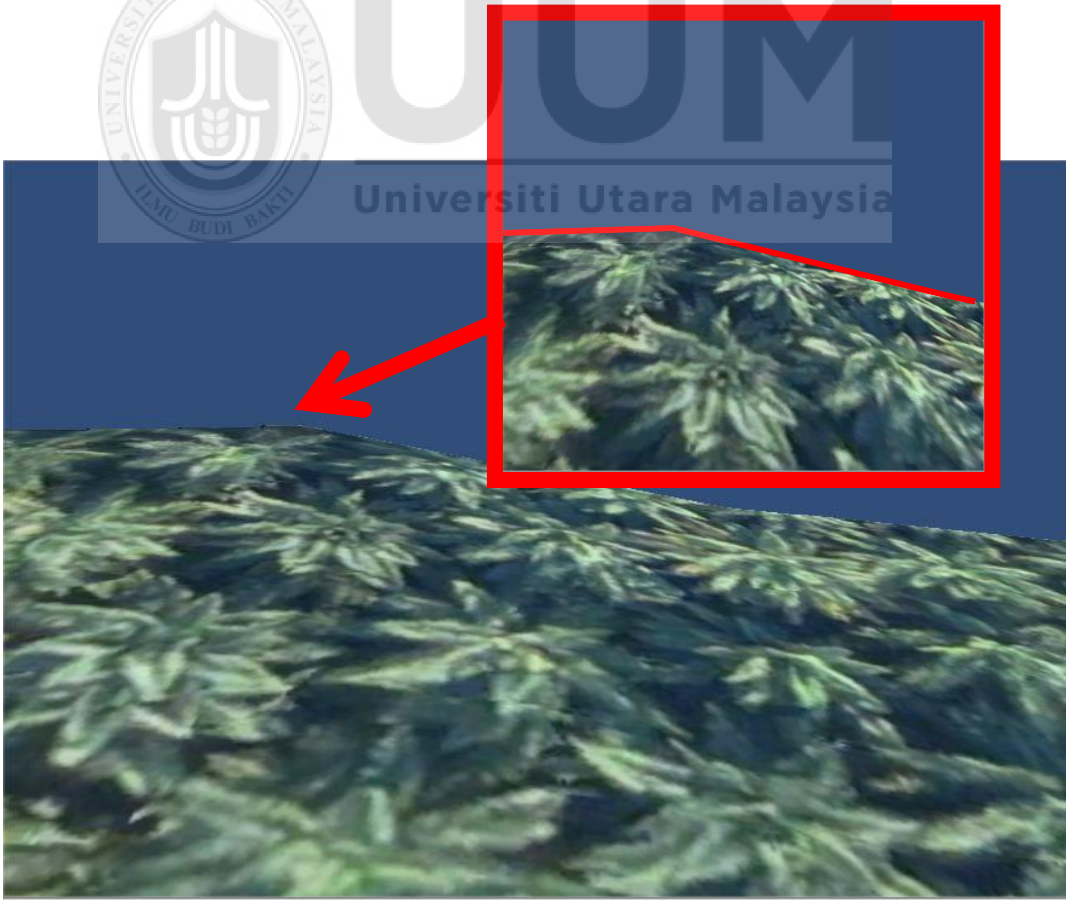

Figure 5.11. The View of UAV Images Draped with 5m Interval Contour Data.

The second result is from 3m interval output. Figure 5.12 shows the image of this output. The results of this image reveal a "Clear" visualization quality where the area inside the circle can be seen clearly but with slight slopes, overall, most of the area inside the oil palm plantation can be viewed clearly.

Figure 5.12. The View of UAV Images Draped with 3m Interval Contour Data.

The final result is from 1m interval output. Figure 5.13 shows the image of this output. The results of this image show a "Very clear" visualization quality where the area inside the circle can be seen very clearly with a good view of slopes, overall, most of the area inside the oil palm plantation can be viewed clearly.

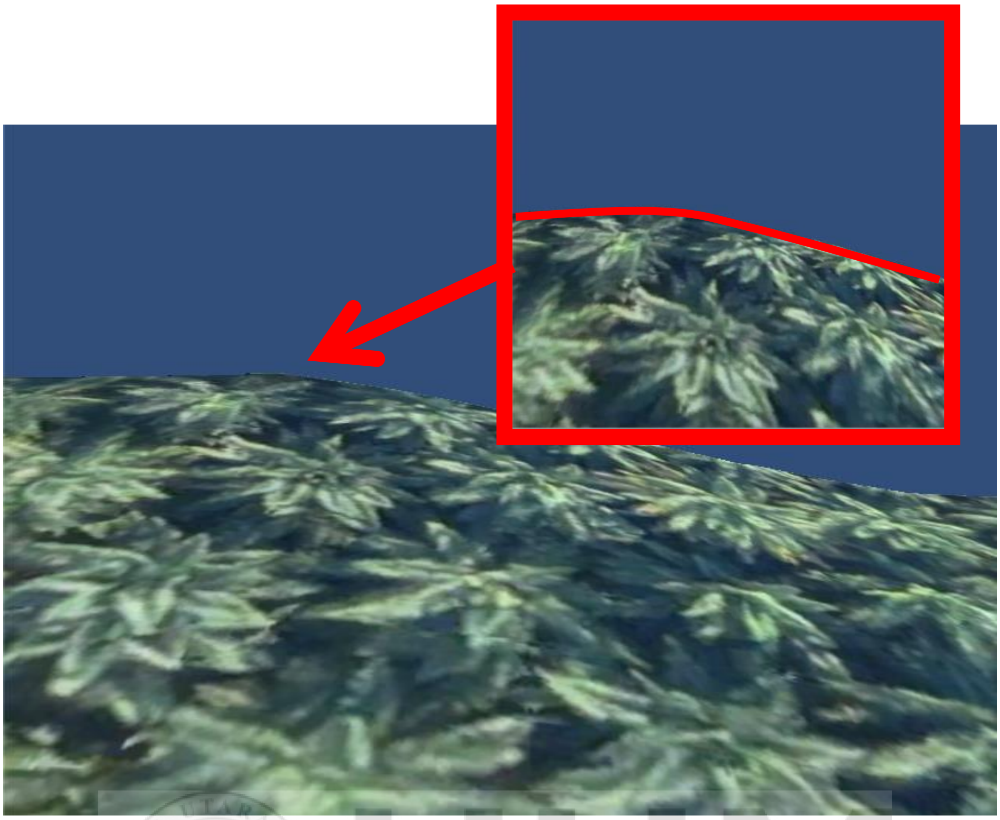

Figure 5.13. The View of UAV Images Draped with 1m Interval Contour Data.

Table 5.7 shows the results produced from each contour interval data. There are five criteria to be measured which are data size, loading time (during office hours and after office hours), response time, visualization quality, and fps. The results of data size for 1m data interval has the bigger file size after published compared to the 3m and 5m data interval. This is due to the fact that 1m data interval usually holds a lot of data compared to the 3m and 5m data interval. The same size of UAV images was overlaid on each data interval. The visualization qualities in these three outputs were already discussed earlier in this section.

Table 5.7

Result of each contours data

Criteria	**Contour Intervals**					
	1m		**3m**		**5m**	
Data size	1.401 kb		1.399 kb		1.399 kb	
Loading time office hour (Off) and after office hour (Out)	Off	Out	Off	Out	Off	Out
	3.2 sec	2.0 sec	1.2 sec	1.3 sec	1.0 sec	0.9 sec
Response time	139 ms		109 ms		110 ms	
Visualization quality	Very clear		Clear		Acceptable	
Frames per second (fps)	58.11 fps		57.54 fps		56.45 fps	

The other criteria to be measured were fps value where Comodo Dragon browser performance tool is used for the purpose of this testing. Testing was done by recording the web page in five seconds for different contour interval to show the value of fps. For 5m data interval, 56.45 fps value was recorded while for 3m data interval 57.54 fps value was recorded and 1m data interval recorded fps value of 58.11. After that, the test continued with measuring the response time for each contour intervals data using Yslow by Duran (2015). In order to view the response time value, the YSlow plug-in is installed inside Mozilla Firefox web browser. The results of the response time acquired are 139 ms for 1m, 109 ms for 3m, and 110 ms for 5m. The response time for each contour interval is slightly different from each other. Finally, the loading time was measured using tools from App.telemery name "page speed monitor" by Fabasoft group (2015). The result that was recorded is from two-time frames that are during office hour and after office hour. The time taken for office-hour was between 8 am to 5 pm while the after office hour was after 5 pm. The results recorded for office-hour are 3.2 sec for 1m, while for 3m and 5m each recorded an average of 1sec. While the results for off office hour is 2 sec for 1m, 1.3 sec for 3m and 0.9 sec for 5m is slightly faster compared to office hour.

In conclusion, it can be concluded that in order to develop online 3D terrain visualization overlaid with UAV images by different contour interval terrain data using Unity 3D game engine, there are some considerations that have to be looked upon. One of it is the accuracy of the data used. By converting the terrain data in multiple formats of data which is raster to vector ArcGIS, the terrain data has extra features when generating terrain in Unity 3D game engine. Thus, the result shows slightly different from the expectation on 1m, 3m, and 5m contour intervals. As for 1m contour interval, it shows that the result as expected where 1m contour interval generated bigger data size compared to 3m contour interval. However, the data size of 5m contour interval is slightly bigger than 3m contour interval. This situation occurred, may be due to the extra features generated from the 5m contour interval. It can be concluded that the most appropriate and suitable data to be used for the implementation of online 3D terrain visualization draped with UAV images using

Unity 3D game engine was 3m interval data. It presented promising results as did the 1m interval data which has less data size, less loading time and response time and less fps value as compared to 1m interval data.

5.4 Summary

Chapter five discussed the result gathered from six different methods that were compared based on loading time, response time, FPS, CPU usage, memory usage and comparison of all measurement for each terrain data size in Unity3D in the online and offline environment. All of these comparisons aimed at measuring the performance of enhanced 3D terrain visualization prototype. The results show that Unity3D file used LZMA file compression which consumes more memory and CPU usage. It can also affect the fps value, response time and loading time because Unity3D have to contact its server to check whether the user has installed the plug-in needed for viewing the terrain. Comparing both online and offline environments, it can be concluded that online environment uses more resources for compression and also forces the browser to request online resources from the operating system. Which is different from an offline environment where the browser is unable to use the online resource from the operating system since it is not active. Overall, the performance of enhancing 3D terrain visualization process prototypes in the offline environment is better than online environment.

CHAPTER SIX
CONCLUSION AND FUTURE WORKS

6.1 Summary of the Research

In summary, the discussion on how to enhance 3D terrain visualization process, developing the prototype based on the enhancing process and evaluating the performance of the enhanced process of 3D terrain visualization using the Unity3D game engine in offline and online environments is presented. The first chapter starts with the introduction of the research and finding the problem statement for the research. The second chapter discussed related literature on related domains, as well as providing information needed to answer objective one. The third chapter discusses how the research going to be implemented and chapter four discuss the development of enhancing 3D terrain visualization process prototype using Unity3D as well fulfilling the second research objectives. Chapter five discusses the results gathered when the testing is executed and finally this chapter discusses the achievement of the research objective from the beginning until the end.

6.2 Achievement of Research Objective

The three objectives have been met successfully which are enhancing 3D terrain visualization process using a game engine, develop the prototype based on enhancing 3D terrain visualization process using a game engine, and to evaluate the performance of enhancing 3D terrain visualization process using game engine.

6.2.1 Enhancing 3D Terrain Visualization Process Using Game Engine

As mentioned earlier in chapter one, the first objective of this study is to enhance 3D terrain visualization process using game engine. Several experiments on different terrain data either using raw data or using the script was reviewed from literature to identify what are the requirements needed for enhancing 3D terrain visualization process using the Unity3D game engine. It was found that the process of enhancing 3D terrain visualization consists of three processes which are terrain data reading, terrain data conversion and finally is terrain data process. The details of each process are explained in Chapter Four.

6.2.2 Prototype Development of Enhancing 3d Terrain Visualization Process Using Game Engine

In this research, the prototype of enhancing 3D terrain visualization process using a game engine that is Unity3D was successfully applied to the online and offline environment. The prototype was developed using the Unity3D game engine. The technologies involved consist of HTTP, Unity3D web player, and web server as well as web browser. The Unity3D web player plug-in is needed for visualizing the terrain in online environments. The prototype consists of terrain view where the user can fly around the terrain to get a better view of the terrain closer. Since it is online, the user can view the terrain from anywhere and at any place as long as internet connection is available. The process of prototype development begins with UAV data collection at RISDA Tanjung Genting plantation, which is located near Universiti Utara Malaysia (UUM). The collected data from the UAV is stitched and further processed to remove the trees to get the raster (DEM) needed for this study. The terrain data is clipped using ArcGIS software and loaded into Unity3D, the script which consists of an algorithm for enhancing 3D terrain visualization process were added and generated to make the process of loading the terrain data onto the Unity3D. After that, the process continues with publishing the system onto the online environment and uploading to the web server.

6.2.3 Evaluation of the Performance of Enhanced Process Of 3D Terrain Visualization Using Game Engine In Offline And Online Environments

The evaluation of the enhancing 3D terrain visualization process prototype is by comparing its performance in two different environments which are online and offline. The analysis of performance is compared based on loading time, response time, frame per second, CPU usage, memory usage and comparing all the measurement for each terrain data size. The results of the comparison between online and offline environment found that the results of loading time in offline is faster compared to online environments. The results for response time also show offline to be faster as compared to online environments. While for the results of FPS value, the FPS value for offline stated higher compared to online environments. Finally, the results for CPU usage and memory usage also revealed that the percentage of CPU usage is higher in offline compared to the results in online environments. While the results of memory usage revealed offline to be lower as compared to online. This situation happens maybe due to Unity3D's need for maximum LZMA compression when publishing the files for web users and also Unity3D locks the file from outside editing. Other than that Unity3D need to check whether the user has installed the web player plug-in for terrain visualization while running the system. Based on all of these results, it can be concluded that the performance of enhancing 3D terrain visualization process using a game engine in the offline environment is better compared to online environments.

Other than that, in order to give an idea for visualization developer to develop web-based 3D terrain visualization using Unity3D, the operational guidelines which consist of experiments on using different size of contour intervals for web-based 3D terrain visualization using Unity3D was conducted. The results of this experiment gave the visualization developer an idea of choosing the suitable types of contour intervals for rendering online 3D terrain visualization using Unity3D.

6.3 Future works

This study has the potential to be extended into different types of applications such as plantation management, security management for flood and forest fire, also in education as an additional study material for school students and teachers. Hence, contributing to the body of knowledge and paving the way for terrain visualization to be widely used.

This study can also be applied to the military sector for planning, mission coordination, and risk evaluation. By having the terrain information at the early stage, it enables the security forces to act faster and reduce resources use to simulate the data in the real world.

This study can also be applied in agriculture for farm planning and finding a suitable area for vegetation replanting. It also can help this sector by generating more food and also preserving forestry as unneeded deforestation can be avoided by having the suitable terrain information.

Finally, this study may also be used in disaster management like flooding, forest fire, and landslides management by providing the necessary terrain information for the rescue forces in case of the occurrence of a natural disaster.

Especially for natural disasters that occur in secluded vicinities as game engines allow users the flexibility to add scenarios.

REFERENCES

Aaron, L. (2013). The top 14 game engines: The list in full. Retrieved June 7 2014 from http://www.develop-online.net/news/the-top-14-game-engines-the-list-in-full/0114330

Agisoft. (2015). Agisoft PhotoScan. Retrieved 20 December 2015 from http://www.agisoft.com/

Alexander, V. (2014). Page load time. Retrieved 10 October 2014 from https://github.com/alex-vv/chrome-load-timer

Analytical Graphics Inc. (2014). Cesium. Retrieved 2 January 2014 from http://cesiumjs.org/index.html

Andre, G. D. (2014). Earth 3D. Retrieved 10 January 2014 from http://www.earth3d.org/index.html

Anroni, Y. E., & Andesti, F. M. (2017). Perancangan Sistem Informasi Pembuatan SIUP di BPMPTSP Kota Padang. *Indonesian Journal of Computer Science, 6*(2), 176-191.

Apple. (2010). What is Frame rate? Retrieved 5 October 2014 from https://documentation.apple.com/en/finalcutpro/usermanual/index.html#chapter=D%26section=1%26tasks=true

ArcGIS Resource Center. (2012). Coordinate systems, map projections, and geographic (datum) transformations. Retrieved 10 September 2016 from http://resources.esri.com/help/9.3/arcgisengine/dotnet/89b720a5-7339-44b0-8b58-0f5bf2843393.htm

Attaway, D. F., Jacobsen, K. H., Falconer, A., Manca, G., Bennett, L. R., & Waters, N. M. (2014). Mosquito habitat and dengue risk potential in Kenya: alternative methods to traditional risk mapping techniques. *Geospatial health, 9*(1), 119-130.

Autodesk. (2013). Creating Animated Bones with Biped. Retrieved 10 October 2014 from http://docs.autodesk.com/3DSMAX/12/ENU/3ds%20Max%202010%20Tutorials/files/WSf742dab04106313311b7d0e3112a19e3350-7fef.htm

Awadallah, R. i. S., Gehman, J. Z., Kuttler, J. R., & Newkirk, M. H. (2004). Modeling Radar Propagation in Three-Dimensional Environments. *Johns Hopkins Apl Technical Digest, 25*, 11.

Baccini, A., Laporte, N., Goetz, S., Sun, M., & Dong, H. (2008). A first map of tropical Africa's above-ground biomass derived from satellite imagery. *Environmental Research Letters, 3*(4), 045011.

Bahor, S., & Ramic-Brkic, B. (2013). HTML5/WebGL vs Flash in 3D Visualisation. *Proceedings of CESCG*.

Berger, M., & Cristie, V. (2015). CFD Post-processing in Unity3D. *Procedia Computer Science, 51*, 2913-2922.

Biosphere. (2007). Biosphere3D: Real-time open source Landscape Visualization. Retrieved 10 January 2014 from http://www.biosphere3d.org/

Bishop, I. D. (2012). On-Line Approaches to Data Delivery and Visualisation in Landscape Planning and Management. *International Journal of E-Planning Research (IJEPR), 1*(1), 31-41.

Bose, M., & Rajagopala, V. (2012). *Physics engine on reconfigurable processor—Low power optimized solution empowering next-generation graphics on embedded platforms.* Paper presented at the 17th International Conference on Computer Games (CGAMES), .

Burdea, G., & Coiffet, P. (2003). Virtual reality technology. *Presence: Teleoperators and virtual environments, 12*(6), 663-664.

Cellier, F., Gandoin, P.-M., Chaine, R., Barbier-Accary, A., & Akkouche, S. (2012). *Simplification and streaming of GIS terrain for web clients.* Paper presented at the Proceedings of the 17th International Conference on 3D Web Technology.

Chang, A., Parrales, M., Jimenez, J., Sobieszczyk, M., Hammer, S., Copenhaver, D., & Kulkarni, R. (2009). Combining Google Earth and GIS mapping technologies in a dengue surveillance system for developing countries. *International Journal of Health Geographics, 8*(1), 49. Retrieved from http://www.ij-healthgeographics.com/content/8/1/49

Chen, H., Peng, R., Li, S., & Yu, C. (2009). *The Visualization of 3D Terrain Based on VRML.* Paper presented at the IFITA '09. International Forum on Information Technology and Applications, 2009.

Chen, H., Wang, Q., Zhao, C., Niu, J., & Zhu, D. (2010). *Implementation of agricultural training system using Game Engine.* Paper presented at the 2nd International Conference on Education Technology and Computer (ICETC), 2010.

Cheung, G. K. M., Kanade, T., Bouguet, J. Y., & Holler, M. (2000). *A real time system for robust 3D voxel reconstruction of human motions.* Paper presented at the IEEE Conference Proceedings on Computer Vision and Pattern Recognition, 2000.

Cole, K. (2012). Creating a Terrain in Unity From a DEM. *CAST Technical Publications Series,*(Number 11449). Retrieved from http://gmv.cast.uark.edu/modeling/software-visualization/unity-software-visualization/workflow-unity-software-visualization/creating-a-terrain-in-unity-from-a-dem/

Collin, L. (2005). A Quick Benchmark: Gzip vs. Bzip2 vs. LZMA. Retrieved 10 October 2016 from http://tukaani.org/lzma/benchmarks.html

Cowgill, E., Bernardin, T. S., Oskin, M. E., Bowles, C., Yıkılmaz, M. B., Kreylos, O., Elliott, A. J., Bishop, S., Gold, R. D., & Morelan, A. (2012). Interactive terrain visualization enables virtual field work during rapid scientific response to the 2010 Haiti earthquake. *Geosphere, 8*(4), 787-804.

Crooks, A. T. (2010). Constructing and implementing an agent-based model of residential segregation through vector GIS. *International Journal of Geographical Information Science, 24*(5), 661-675.

CryTEK. (2014). Cry Engine 3. Retrieved 12 January 2014 from http://www.crytek.com/cryengine/cryengine3/

Dar-Hsiung, W., Han-Ching, H., Chin-Shien, W., Honjo, T., Yu-Jen, C., & Pin-An, Y. (2012). *Visualization with Google Earth and gaming engine.* Paper presented at the IEEE Fourth International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications (PMA), 2012

Davis, W. S. (1998). HIPO hierarchy plus input-process-output. *The information system consultant's handbook: Systems Analysis and Design. CRC, Florida*, 503-510.
Dübel, S., & Schumann, H. (2017). Visualization of Features in 3D Terrain. *ISPRS International Journal of Geo-Information, 6*(11), 357.
Duran, M. (2015). yslow. Retrieved 30 December 2015 from yslow.org
Encylopedia.com. (2014). Pierre de Fermat. Retrieved August 4 2014 from http://www.encyclopedia.com/topic/Pierre_de_Fermat.aspx
Endres, M. (2017). Creating a level using World Machine Tutorial. Retrieved 10 August2017 from http://docs.cryengine.com/display/SDKDOC2/Creating+a+level+using+World+Machine+Tutorial
Environmental Systems Research Institute (ESRI). (2015). ArcGIS. Retrieved 10 May 2015 from https://www.arcgis.com/features/index.html
EPA. (2014). What is a Geographic Information System (GIS)? Retrieved 10 October 2014 from http://www.epa.gov/airmarkets/maps/what.html
Epic Games, I. (2012). Creating New Landscapes Retrieved 10 August 2017 from https://docs.unrealengine.com/udk/Three/LandscapeCreating.html
Epic Games, I. (2014). Unreal Development Kit. Retrieved 10, January 2014 from http://www.unrealengine.com/udk/downloads/
ESRI. (2014). What is GIS? Retrieved August 4 2014 from http://www.esri.com/what-is-gis/howgisworks
ESRI Press. (2005). Charting the Unknown. Retrieved 10 October 2014 from http://www.gsd.harvard.edu/gis/manual/lcgsa/HarvardBLAD_screen.pdf
Fabasoft. (2014). Page Speed Monitor. Retrieved 10 October 2014 from http://www.apptelemetry.com/en/page-speed-monitor.html
Fabasoft Group. (2015). app.telemetry. Retrieved 12 December 2015 from http://www.apptelemetry.com/index_en.html
Flora3D. (2014). Flora3D. Retrieved 10 October 2014 from http://flora3d.net/
Friese, K.-I., Herrlich, M., & Wolter, F.-E. (2008). Using game engines for visualization in scientific applications *New Frontiers for Entertainment Computing* (pp. 11-22): Springer.
Fu, P., Sun, J., & Yin, F. (2011). GIS in the web era. *Web GIS: Principles and applications*, 2-24.
GarageGames.com. (2014). Torque game engine. Retrieved 12 January 2014 from http://www.garagegames.com/products/torque-3d/overview
Gary Sherman. (2004). QGIS. Retrieved 10 June 2016 from http://qgis.org/en/site/
Geodetic Institute of Slovenia. (2015). Data acquisition and processing. Retrieved 20 December 2015 from http://www.gis.si/en/storitve/zajem-in-obdelava-podatkov
Geomantics. (2014). Genesis IV. Retrieved 10 January 2014 from http://www.geomantics.com/genesis4.htm
GISGeography (2016). GIS Spatial Data Types: Vector vs Raster Retrieved 20 October 2016 from http://gisgeography.com/spatial-data-types-vector-raster/
González, C., Pérez, M., & Orduña, J. M. (2016). Combining displacement mapping methods on the GPU for real-time terrain visualization. *The Journal of Supercomputing*, 1-12.
Google.inc. (2014). Google earth. Retrieved 10 October 2014 from https://www.google.com/earth/explore/products/mobile.html
Gregory, J. (2009). *Game Engine Architecture*: Taylor & Francis.

Guth, P. (2014). MicroDEM. Retrieved 20 January 2014 from http://www.usna.edu/Users/oceano/pguth/website/microdem/microdem.htm

gvSIG Association. (2015). gVSIG. Retrieved 10 June 2016 from http://www.gvsig.com

Hagedorn, B., & Döllner, J. (2007). *High-level web service for 3D building information visualization and analysis.* Paper presented at the Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems.

Hayat, K., Puech, W., & Gesquiere, G. (2008). Scalable 3-D Terrain Visualization Through Reversible JPEG2000-Based Blind Data Hiding. *IEEE Transactions on Multimedia, 10*(7), 1261-1276. doi:10.1109/tmm.2008.2004905

Herman, B. (2016). Quick Benchmark: Gzip vs Bzip2 vs LZMA vs XZ vs LZ4 vs LZO. Retrieved 11 November 2016 from https://catchchallenger.first-world.info/wiki/Quick_Benchmark:_Gzip_vs_Bzip2_vs_LZMA_vs_XZ_vs_LZ4_vs_LZO

Hodges, W. (2001). Logic and Games. *2014*(August 4). Retrieved from http://plato.stanford.edu/entries/logic-games/

Hoffman, C. (2014). HTG Explains: Why It's Good That Your Computer's RAM Is Full. Retrieved 10 October 2014 from http://www.howtogeek.com/128130/htg-explains-why-its-good-that-your-computers-ram-is-full/

Horne, R. (2014). 3DEM. Retrieved 10 January 2014 from http://freegeographytools.com/2009/3dem-website-is-gone-but-3dem-still-available-here

Huisman, O., & By, R. A. d. (Eds.). (2009). *Principles of Geographic Information Systems: an introductory book* (Fourth ed.): The International Institute for Geo-Information Science and Earth Observation.

Humbert, P., Chevrier, C., & Bur, D. (2011). Use Of A Real Time 3D Engine For The Visualization Of A Town Scale Model Dating From The 19th Century. *Prague, République tchèque*.

Hussain, S., Wang, Z., Toure, I. K., & Diop, A. (2013). Web service testing tools: A comparative study. *arXiv preprint arXiv:1306.4063*.

İncekara, S. (2012). *Do Geographic Information Systems (GIS) Move High School Geography Education Forward in Turkey? A Teacher's Perspective*.

Indraprastha, A., & Shinozaki, M. (2009). The investigation on using Unity3D game engine in urban design study. *Journal of ICT Research and Applications, 3*(1), 1-18.

Ismaeel, A. G., & Hamead, N. G. (2014). Mobile GIS and Open Source Platform Based on Android: Technology for System Pregnant Women. *arXiv preprint arXiv:1402.6282*.

Jacob, S. (2016). Speed Is A Killer – Why Decreasing Page Load Time Can Drastically Increase Conversions. Retrieved 10 August 2017 from https://blog.kissmetrics.com/speed-is-a-killer/

Jajac, N., Stojanovic, D., Predic, B., & Rancic, D. (2013). *Efficient replication of geospatial data for mobile GIS in field work.* Paper presented at the 11th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS), 2013

Janus Research Group Inc. (2014). Janus Research. Retrieved 4 August 2014 from http://www.janusresearch.com

Janzen, B. F., & Teather, R. J. (2014). *Is 60 FPS better than 30?: the impact of frame rate and latency on moving target selection.* Paper presented at the Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems.

Japaridze, G. (2014). Computability Logic Homepage. Retrieved 4 August 2014 from http://www.cis.upenn.edu/~giorgi/cl.html

Jarvis, C., Løvset, T., & Patel, D. (2015). *Revisiting virtual reality training using modern head mounted display and game engines.* Paper presented at the Proceedings of the 8th International Conference on Simulation Tools and Techniques.

Juarez, A., Schonenberg, W., & Bartneck, C. (2010). Implementing a low-cost CAVE system using the CryEngine2. *Entertainment Computing, 1*(3), 157-164.

Kang, Y., Kim, H., & Han, S. (2015). Visualization of the Synthetic Environment Data Representation & Interchange Specification data for verifying large-scale synthetic environment data. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, 12*(4), 507-518.

Karakuyu, M. (2010). How are Geographic Information Systems (GIS) conferences at Fatih University contributing to education in Turkey? *Scientific Research and Essays, 5*(19), 2975-2982.

Klopfer, E., Osterweil, S., Groff, J., & Haas, J. (2009). Using the technology of today, in the classroom today. The instructional power of digital games, social networking, simulations, and how teachers can leverage them. *The Education Arcade*, 1-21.

Kumar, S., Chhugani, J., Kim, C., Kim, D., Nguyen, A., Dubey, P., Bienia, C., & Kim, Y. (2008). Second life and the new generation of virtual worlds. *Computer, 41*(9).

Latifoski, E., Kotevski, Z., & Hristoski, I. (2016). Visualization of flood data using HTML5 technologies.

Lenne3D. (2014). Lenne3D. Retrieved 10 October 2014 from http://lenne3d.com/

Lewis, M., & Jacobson, J. (2002). Introduction. *Commun. ACM, 45*(1), 27-31. doi:10.1145/502269.502288

Low, D. (2017). 10 Frustrating Reasons Why Your Website Speed Is Slow. Retrieved 10 August 2017 from https://www.bitcatcha.com/blog/2015/10-frustrating-reasons-why-your-website-speed-is-slow/

Maa, Q., Yangb, Z., Chen, H., Zhu, D., & Guo, H. (2012). A Serious Game for Teaching and Learning Agricultural Machinery Driving.

Martin, H., Chevallier, S., & Monacelli, E. (2016). Adaptive Visualisation System for Construction Building Information Models Using Saliency. *arXiv preprint arXiv:1603.02028*.

Mazuryk, T., & Gervautz, M. (1996). Virtual reality-history, applications, technology and future.

Messaoudi, F., Simon, G., & Ksentini, A. (2015). *Dissecting games engines: The case of Unity3D.* Paper presented at the Network and Systems Support for Games (NetGames), 2015 International Workshop on.

Michaelenger. (2013). Game Engines: How do they work? Retrieved 10 October 2014 from http://www.giantbomb.com/profile/michaelenger/blog/game-engines-how-do-they-work/101529/

Microsoft. (2014). Process Explorer. Retrieved 10 October 2014 from https://technet.microsoft.com/en-us/sysinternals/bb896653.aspx

Microsoft. (2017). Networking and Internet. Retrieved 10 August 2017 from https://msdn.microsoft.com/en-us/library/windows/desktop/ee663286(v=vs.85).aspx

Moore, I. D., Gessler, P. E., Nielsen, G. A., & Peterson, G. A. (1993). Soil Attribute Prediction Using Terrain Analysis. *Soil Science Society of America Journal, 57*(2), 443-452. doi:10.2136/sssaj1993.03615995005700020026x

Morgan, J. L., Gergel, S. E., & Coops, N. C. (2010). Aerial photography: a rapidly evolving tool for ecological management. *BioScience, 60*(1), 47-59.

Natarajan, P., Amer, P. D., & Stewart, R. (2008). *Multistreamed web transport for developing regions*. Paper presented at the Proceedings of the second ACM SIGCOMM workshop on Networked systems for developing regions, Seattle, WA, USA.

National Geographic Society. (2014). GIS (Geographic Information System). Retrieved 15 April 2014 from http://education.nationalgeographic.com/education/encyclopedia/geographic-information-system-gis/?ar_a=1

Navarro, A., Pradilla, J. V., & Rios, O. (2012). Open Source 3D Game Engines for Serious Games Modeling. *Modeling and Simulation in Engineering*. Retrieved from http://www.intechopen.com/books/modeling-and-simulation-in-engineering/open-source-3d-game-engines-for-serious-games-modeling

Ningsih, S. (2017). Sistem Informasi Parkir INAP Mobil Pada PT. Angkasa Pura II Solusi. *JOISIE (Journal Of Information Systems And Informatics Engineering), 5*(2), 84-90.

Norkart. (2014). Norkart Virtual Globe. Retrieved 20 January 2014 from http://www.virtual-globe.info

OpenEV. (2014). OpenEV. Retrieved 20 January 2014 from http://openev.sourceforge.net/index.php?page=features

Oxendine, C. E., & Waters, N. (2014). No-Notice Urban Evacuations: Using Crowdsourced Mobile Data to Minimize Risk. *Geography Compass, 8*(1), 49-62.

Oxford University Press. (2005). Cartogram. Retrieved 4 August 2014 from http://www.oxforddictionaries.com/definition/english/cartogram?q=cartograms

Oxford University Press. (2014). Choropleth Map. Retrieved 4 August 2014 from http://www.oxforddictionaries.com/definition/english/choropleth-map?q=Choropleth+maps

PCI Geomatics. (2015). Geomatica. Retrieved 20 December 2015 from http://www.pcigeomatics.com/

Philipp Offermann, Olga Levina, Marten Schnherr, & Udo Bub. (2009). *Outline of a design science research process*. Paper presented at the Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, Philadelphia, Pennsylvania.

Planetside Software. (2016). Terragen 4 download. Retrieved 10 October 2016 from https://planetside.co.uk/free-downloads/terragen-4-free-download/

Pranckevicius, A. N. (2014). FramesPerSecond. Retrieved 10 October 2014 from http://wiki.unity3d.com/index.php?title=FramesPerSecond

Prasithsangaree, P., Manojlovich, J. M., Chen, J., & Lewis, M. (2003). *UTSAF: a simulation bridge between OneSAF and the Unreal game engine*. Paper presented at the IEEE International Conference on Systems, Man and Cybernetics, 2003.

Raber, G. T., Jensen, J. R., Schill, S. R., & Schuckman, K. (2002). Creation of digital terrain models using an adaptive lidar vegetation point removal process. *Photogrammetric engineering and remote sensing, 68*(12), 1307-1314.

Rathnam, R., Pfingsthorn, M., & Birk, A. (2009, 3-6 Nov. 2009). *Incorporating large scale SSRR scenarios into the high fidelity simulator USARSim.* Paper presented at the IEEE International Workshop on Safety, Security & Rescue Robotics (SSRR), 2009

Real Visual Group. (2014). Real Visual. Retrieved 10 October 2014 from http://www.realvisualgroup.com/

Roccatello, E., Nozzi, A., & Rumor, M. (2013). Design and development of a framework based on OGC web services for the visualization of three dimensional large-scale geospatial data over the web. *ISPRS Archive, XL-4/W1*, 101-104.

Rodgers, C. (2014). *Augmented Reality Books and the Reading Motivation of Fourth-Grade Students*: Union University.

Röhlig, M., & Schumann, H. (2016). *Visibility Widgets: Managing Occlusion of Quantitative Data in 3D Terrain Visualization.* Paper presented at the Proceedings of the 9th International Symposium on Visual Information Communication and Interaction.

Rutgers. (2017). Challenges and Benefits of Information Visualization - Master of Information. Retrieved 10 August 2017 from http://online.rutgers.edu/resources/articles/challenges-and-benefits-of-information-visualization-master-of-information/?program=mi

Ruzinoor, C. M., Abdul Rashid , M. S., Pradhan, B., Ahmad Rodzi , M., & Mohd Shafry, M. R. (2013). An effective visualization and comparison of online terrain draped with multi-sensor satellite images. *Arabian Journal of Geosciences, 6*(12), 4881-4889.

Ruzinoor, C. M., Mohamed Shariff, A. R., Pradhan, B., & Ahmad Rodzi, M. (2011). Online 3D terrain visualization of GIS data: A comparison between three different web servers. *Pertanika Journal of Science & Technology, 19*(S), 31-39.

Ruzinoor, C. M., Sharif, A. R. M., Mahmud, A. R., & Pradhan, B. (2008). *Online 3D Real Time Terrain Rendering Algorithm for GIS Data: A Conceptual Idea.* Paper presented at the 3rd International Workshop on 3D Geoinformation 2008, Seoul, South Korea

Ruzinoor, C. M., Shariff, A. R. M., & Mahmud, A. R. (2009). *Online 3D terrain visualization: A comparison of three different GIS software.* Paper presented at the International Conference on Information Management and Engineering, 2009. ICIME'09.

Ruzínoor, C. M., Shariff, A. R. M., Mahmud, A. R., & Pradhan, B. (2012). 3D Terrain Visualisation for GIS: A Comparison of Different Techniques. In M. Buchroithner (Ed.), *True-3D in Cartography* (pp. 265-277): Springer Berlin Heidelberg.

Ruzinoor, C. M., Shariff, A. R. M., Pradhan, B., Rodzi Ahmad, M., & Rahim, M. S. M. (2012). A review on 3D terrain visualization of GIS data: Techniques and software. *Geo-spatial Information Science, 15*(2), 105-115. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-84880972430&partnerID=40&md5=a9f30ce768d684033a7c3e2229687023

Santos, M. E. C., Chen, A., Taketomi, T., Yamamoto, G., Miyazaki, J., & Kato, H. (2014). Augmented reality learning experiences: Survey of prototype design and evaluation. *IEEE Transactions on learning technologies, 7*(1), 38-56.

SAS Institute Inc. (2017). Data Visualization. Retrieved 10 August 2017 from https://www.sas.com/en_us/insights/big-data/data-visualization.html

SEOchat. (2017). Average Page Load Time of Top Ranking Websites in Google. Retrieved 10 August 2017 from http://www.seochat.com/c/a/google-optimization-help/average-page-load-time-of-top-ranking-websites-in-google/

sgerhardt. (2011). Projected Coordinate System vs. Geographic Coordinate System. Retrieved 11 October 2016 from https://geonet.esri.com/thread/23160

Sherif, M., & Abdul-Kader, H. (2011). Novel robust multilevel 3D visualization technique for web based GIS. *The International Arab Journal of Information Technology,, 8*(1), 59-65.

Sherman, W. R., & Craig, A. B. (2003). *Understanding Virtual Reality: Interface, Application, and Design*: Morgan Kaufmann.

Shin, I.-S., Beirami, M., Cho, S.-J., & Yu, Y.-H. (2015). Development of 3D Terrain Visualization for Navigation Simulation using a Unity 3D Development Tool. *Journal of the Korean Society of Marine Engineering, 39*(5), 570-576.

Skemer, D. (2013). Berthier's Manuscript Maps of America, 1781–82. Retrieved 4 August 2014 from https://blogs.princeton.edu/manuscripts/2013/11/27/berthiers-manuscript-maps-of-america-1781-82/

Smith, M. (2014). How To Fix High CPU Usage In Windows. Retrieved 10 October 2014 from http://www.makeuseof.com/tag/fix-high-cpu-usage-windows/

Stenneth, L., Wolfson, O., Yu, P. S., & Xu, B. (2011). *Transportation mode detection using mobile phones and GIS information.* Paper presented at the Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.

Stock, M. J. (1998). Hftools. Retrieved 15 January 2014 from http://markjstock.org/pages/hftools.html

Supergeo. (2006). Ancient Map Geographic Information System. Retrieved 10 August 2016 from http://www.supergeotek.com/inner/Download/Ancient%20Map%20Geographic%20Information%20System_201004061-20100406180148.pdf

Systemics Blog. (2013). Cartesian coordinate system. Retrieved 10 October 2014 from http://www.systemicsblog.com/en/2011/cartesian_coordinate_system/

Tateyama, Y., Oonuki, S., Sato, S., & Ogi, T. (2008). *K-Cave demonstration: Seismic information visualization system using the OpenCABIN library.* Paper presented at the Proceedings of the ICAT.

The University of Washington Spatial Technology GIS and Remote Sensing. (2013). The GIS Spatial Data Model. Retrieved 12 November 2016 from https://courses.washington.edu/gis250/lessons/introduction_gis/spatial_data_model.html

The Urban and Regional Information Systems Association. (2013). Roger Tomlinson. Retrieved 10 October 2014 from http://www.urisa.org/awards/roger-tomlinson/

Tsou, M. H. (2004). Integrated mobile GIS and wireless internet map servers for environmental monitoring and management. *Cartography and Geographic Information Science, 31*(3), 153-165.

TutorialsPoint. (2017). Software Analysis & Design Tools. Retrieved 10 August 2017 from https://www.tutorialspoint.com/software_engineering/software_analysis_design_tools.htm

UCGIS. (2014). Roger Tomlinson. Retrieved 7 June 2014 from http://ucgis.org/ucgis-fellow/roger-tomlinson

Unidata. (2014). Integrated Data Viewer (IDV). Retrieved 12 January 2014 from http://www.unidata.ucar.edu/software/idv/

United States Department of Agriculture (USDA). (2012). Percent of Farms with High-Speed Internet Access. Retrieved 12 October 2015 from https://www.agcensus.usda.gov/Publications/2007/Online_Highlights/Ag_Atlas_Maps/Farms/Number/07-M252.php

Unity3D. (2014). Unity3D. Retrieved 10 January 2014 from http://unity3d.com/

Utah Department of Environmental Quality (UDEQ). (2015). DEQ: Utah Mapping Partnership Will Increase Government Efficiency, Transparency Retrieved 15 December 2015 from http://dequtah.blogspot.my/2015/03/deq-utah-mapping-partnership-will.html

Vasiljević, T. Z., Srdjević, Z., Bajčetić, R., & Miloradov, M. V. (2012). GIS and the analytic hierarchy process for regional landfill site selection in transitional countries: a case study from Serbia. *Environmental Management, 49*(2), 445-458.

Veronesi, F., & Hurni, L. (2015). A GIS tool to increase the visual quality of relief shading by automatically changing the light direction. *Computers & Geosciences, 74*, 121-127.

Wang, H., Liu, M., Guo, Y., & Chen, X. (2014). *Similarity-based web browser optimization*. Paper presented at the Proceedings of the 23rd international conference on World wide web, Seoul, Korea.

Wang, P., Li, J., & Zhang, Y. (2014). Guaranteed Cost Nonfragile Robust Controller for Walking Simulation of Quadruped Search Robot on a Slope of VRML Model. *Advances in Mechanical Engineering, 6*, 948795.

Wang, S., Mao, Z., Zeng, C., Gong, H., Li, S., & Chen, B. (2010). *A new method of virtual reality based on Unity3D*. Paper presented at the Geoinformatics 2010 18th International Conference, Beijing University, Beijing, China.

Wang, X., Xuedong, Y., Jiangfeng, W., & Dan, L. (2012). Road Scene Modeling for Driving Simulator Based on Tile Library Concept. *Information Technology Journal, 11*(4), 466.

Wood, J. (2014). LandSerf. Retrieved 20 January 2014 from http://www.landserf.org/

Wyeld, T. G., Carroll, J., Ledwich, B., Leavy, B., Gibbons, C., & Hills, J. (2007). *The ethics of indigenous storytelling: using the torque game engine to support Australian Aboriginal cultural heritage.* Paper presented at the Proceedings of DiGRA Conference.

Wyld, D. C. (2010). The virtual tourist: using the virtual world to promote the real one. . *Advances in Competitiveness Research, 18*.

Xu, C., Xu, X., Dai, F., Xiao, J., Tan, X., & Yuan, R. (2012). Landslide hazard mapping using GIS and weight of evidence model in Qingshui River watershed of 2008 Wenchuan earthquake struck region. *Journal of Earth Science, 23*(1), 97-120. doi:10.1007/s12583-012-0236-7

Yang, T., Wuensche, B. C., & Lobb, R. J. (2004). *Game Engine Support for Terrain Rendering in Architectural Design.* Computer Science)--University of Auckland.

Yildirim, V. (2012). Application of raster-based GIS techniques in the siting of landfills in Trabzon Province, Turkey: a case study. *Waste Management & Research*, *30*(9), 949-960.

Yusuf, M. A., Mostafa, M. G., & Elarif, T. I. (2014). A Multi-threaded Multi-context GPU Methodology for Real-time Terrain Rendering.

Appendixes

Appendix A

Heightmap To Gridfloat unity script

```
import System;
import System.IO;

@MenuItem ("Terrain/Heightmap From GridFloat")



static function ApplyHeightmap () {

	var cellSize : float = 10;//adjust this according to the resolution of the file you
are using.

	var filepath : String = null;
	var hdrpath : String = null;

	var bytes : byte[];
	var newHeights : Array[];
	var width = 0;//number of columns of GridFloat data
	var height = 0; //number of rows of GridFloat data
	var maxHeight = 0.0; //max height value in the file

	filepath = EditorUtility.GetAssetPath(Selection.activeObject) ;
	hdrpath = Path.ChangeExtension(filepath,".hdr") ;
if (filepath == null) {
EditorUtility.DisplayDialog("No file selected", "Please select a FloatGrid (.flt) file.",
"Cancel") ;
return;
}


	if(File.Exists(hdrpath) ) {
		var hdrReader : StreamReader = new StreamReader(hdrpath) ;
		var hdrTemp : String = null;
		hdrTemp = hdrReader.ReadLine() ;
		while(hdrTemp!=null) {
			var spaceStart : int = hdrTemp.IndexOf(" ") ;
			var spaceEnd : int = hdrTemp.LastIndexOf(" ") ;

			hdrTemp=hdrTemp.Remove(spaceStart, spaceEnd-spaceStart)
;

			var lineTemp : String[] = hdrTemp.Split(" "[0]) ;
```

```
                        switch(lineTemp[0]) {
                                case "nrows":
                                        height = Int32.Parse(lineTemp[1]) ;
                                        break;
                                case "ncols":
                                        width = Int32.Parse(lineTemp[1]) ;
                                        break;
                                default:
                                        break;
                        }
                        hdrTemp = hdrReader.ReadLine() ;
                }
        }
        else{
                EditorUtility.DisplayDialog("File not found!", "The header (HDR)
file is missing.", "Cancel") ;
return;
        }

        if(File.Exists(filepath) ) {
                bytes = File.ReadAllBytes(filepath) ;
                newHeights= new Array[height];

                for(var i=0; i<height;i++) {
                        newHeights[i]=new Array[width];
                        for(var j=0; j<width; j++) {
                                newHeights[i][j]                                        =
BitConverter.ToSingle(bytes,i*width*4+j*4) ;
                                if(newHeights[i][j]>maxHeight)        maxHeight        =
newHeights[i][j];
                        }
                }
        }
        else{
                EditorUtility.DisplayDialog("File not found!", "Odd, I thought I saw
it.", "Cancel") ;
return;
        }


if (!EditorUtility.DisplayDialog("Warning", "This were to overwrite the existing
heightmap; no undo is possible. Are you sure you want to proceed?", "Apply
heightmap", "Cancel") ) {
return;
}

var terrain = Terrain.activeTerrain.terrainData;

var w2 = terrain.heightmapWidth;
var heightmapData = terrain.GetHeights(0, 0, w2, w2) ;
```

```
var wRatio : float = (width*1.0) /(w2*1.0) ;
var hRatio : float = (height*1.0) /(w2*1.0) ;

terrain.size.x = Mathf.Floor(width*cellSize) ;
terrain.size.z = Mathf.Floor(height*cellSize) ;
terrain.size.y=maxHeight;


for (y = 0; y < w2; y++) {
for (x = 0; x < w2; x++) {

            var tempU = 0;
            var tempD = 0;
            var tempL = 0;
            var tempR = 0;
            var tempUL = 0;
            var tempUR = 0;
            var tempDL = 0;
            var tempDR = 0;

            if(Mathf.Floor(x*wRatio) >0 && Mathf.Floor(x*wRatio) <width-1 && Mathf.Floor(y*hRatio) >0 && Mathf.Floor(y*hRatio) <height-1) {
                tempL = (newHeights[Mathf.Floor(y*hRatio) ][Mathf.Floor(x*wRatio) -1] - newHeights[Mathf.Floor(y*hRatio) ][Mathf.Floor(x*wRatio) ]) *wRatio;
                tempR = (newHeights[Mathf.Floor(y*hRatio) ][Mathf.Floor(x*wRatio) ] - newHeights[Mathf.Floor(y*hRatio) ][Mathf.Floor(x*wRatio) +1]) *wRatio;
                tempU = (newHeights[Mathf.Floor(y*hRatio) -1][Mathf.Floor(x*wRatio) ] - newHeights[Mathf.Floor(y*hRatio) ][Mathf.Floor(x*wRatio) ]) *hRatio;
                tempD = (newHeights[Mathf.Floor(y*hRatio) ][Mathf.Floor(x*wRatio) ] - newHeights[Mathf.Floor(y*hRatio) +1][Mathf.Floor(x*wRatio) ]) *hRatio;
            }

            var avg = (newHeights[y*hRatio][x*wRatio]) + (tempL+tempR+tempU+tempD) ;
            heightmapData[x,y] = avg/terrain.size.y;
}
}
terrain.SetHeights(0, 0, heightmapData) ;
}
```