UNIVERSITY OF CAPE TOWN

MASTERS DISSERTATION

DESIGN AND CREATION OF A VIRTUAL WORLD OF PETRA, JORDAN

STEPHEN WESSELS

2015

SUPERVISOR:

PROF. DR. HEINZ RÜTHER

University of Cape Town

Plagiarism Declaration

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own.

Cape Town, 16 February, 2015

Abstract

This thesis presents the design and creation of a 3D virtual world of Petra, Jordan, based on the digital spatial documentation of this UNESCO World Heritage Site by the Zamani project.

Creating digital records of the spatial domain of heritage sites is a well-established practice that employs the technologies of laser scanning, GPS and traditional surveys, aerial and close range photogrammetry, and 360-degree panorama photography to capture spatial data of a site. Processing this data to produce textured 3D models, sections elevations, GISs, and panorama tours to has led to the establishment of the field of virtual heritage.

Applications to view this spatial data are considered too specialised to be used by the general public with only trained heritage practitioners being able to use the data. Additionally, data viewing platforms have not been designed to allow for the viewing of combinations of 3D data in an intuitive and engaging manner as currently each spatial data type must be viewed by independent software. Therefore a fully integrated software platform is needed which would allow any interested person, without prior training, easy access to a combination of spatial data, from anywhere in the world.

This study seeks to provide a solution to the above requirement by using a game engine to assimilate spatial data of heritage sites in a 3D virtual environment where a virtual visitor is able to interactively engage with combinations of spatial data. The study first begins with an analysis of what virtual heritage applications, in the form of virtual environments, have been created, and the elements that were used in their creation. These elements are then applied to the design and creation of the virtual world of Petra.

Acknowledgements

I would like to express my gratitude to my supervisor, Prof. Heinz Rüther, for his guidance and support during this research. I would also like to thank my co-workers at the Zamani project Roshan Bhurtha and Ralph Schroeder as well as Christoph Held, a former Zamani team member.

This research was hosted by the Zamani project which is based in the Geomatics Division at the University of Cape Town.

A further thank you goes to the UNESCO Staff in Amman, colleagues from the PAP and the Jordanian Department of Antiquities‘ and the Petra National Trust. I am especially thankful to UNESCO for giving the Zamani team the unique opportunity to document Petra, one of the most fascinating and important heritage sites in the world.

Finally I would like to say thank you to my parents, sister and friends for supporting and encouraging me.

Contents

List of Figures

Abbreviations

AR – Augmented Reality

CPU – Central Processing Unit

DOA - Department of Antiquities

DTM – Digital terrain model

FPC – First Person Controller

GUI – Graphical User Interface

ISPRA - The Institute for Environmental Protection and Research

PAP – Petra Archaeological Park

PC – Personal Computer

PNT – Petra National Trust

SfM – Structure from Motion

SG – Serious Game

UNESCO – United Nations Education Science Culture

VH – Virtual Heritage

VR – Virtual Reality

VRML – Virtual Reality Modeling Language

VW – Virtual World

Chapter 1

Introduction

The Zamani project, based in the Geomatics division at the University of Cape Town (UCT), was established by Prof. Heinz Rüther in 2004 to document heritage sites in Africa. To date 40 sites in 15 African and Middle Eastern countries have been recorded to generate digital spatial materials that populate the African Heritage Sites and Landscapes Database. This data has been well received by conservation experts, site managers and researchers who have used the data to manage sites, carry out conservation interventions and conduct research. However, the general public generally has limited use of and access to the spatial heritage data without specialised knowledge and training (Stanco et al., 2011).

The gaming and movie industries have advanced significantly in displaying and interacting with 3D content and thus audiences have been exposed to cutting-edge virtual reality systems. This exposure has developed an expectation of what is possible in a virtual heritage environment. The field of virtual heritage is thus trying to address a growing need for interactive 3D content of heritage sites by creating virtual heritage applications that use digital spatial data to allow the public to virtually interact with sites and artefacts. Applications in the heritage domain, however, lag behind other mainstream industries due to a number of reasons explored in this thesis.

With the objectives of accessibility and interactivity in mind this research was undertaken to create an immersive virtual environment, commonly known as a virtual world, of a cultural heritage site by using a game engine. In this virtual world a virtual visitor is then able to interact with different types of spatial data, without prior training or specialised knowledge of the site, in one seamless and easy to use platform, online and from anywhere in the world.

The World Heritage Site of Petra, Jordan, was chosen for this research. Petra is a 2000 year old Nabataean/Roman site that consists of a 1.2km long access canyon, known as the Siq, a number of adjoining valleys with tombs and temples carved from the sandstone terrain, and free-standing structures. The Zamani project was contracted by UNESCO, in partnership with the Jordanian Department of Antiquities (DOA) and ISPRA (The Institute for Environmental Protection and Research) under the framework of the ―Petra Siq Stability project", to create spatial data of the site for conservation and digital preservation. These data were to be combined into a virtual world of Petra which enables a virtual visitor to virtually walk through and experience the site as if they were there in person. The virtual world is designed to be accessible online or as a stand-alone application usable on standard PC's. The virtual world was created using the Unity game engine software.

Chapter 2

Background

This chapter provides a brief description of cultural heritage and the spatial digital documentation thereof and is followed by a discussion on virtual heritage and its relevance in today‘s world. A description of the author‘s ideal of a virtual world is then given followed by this research‘s objectives, research questions and limitations.

2.1 Cultural heritage

A standardised definition of cultural heritage has not been explicitly defined and adopted internationally as no uniformity exists between the views of individuals, organisations and countries (Ahmad, 2006). The International Council of Monuments and Sites (ICOMOS) describes cultural heritage as ―an expression of the ways of living developed by a community and passed on from generation to generation, including customs, practices, places, objects, artistic expressions and values. Cultural heritage is often expressed as either Intangible or Tangible Cultural Heritage" (ICOMOS, 2002).

Traditionally cultural heritage or just heritage was defined as being historical monuments and settings (Venice Charter, 1964). The scope of cultural heritage was more clearly defined by UNESCO and ICOMOS to include monuments, groups of buildings and sites as well as natural heritage (Ahmad, 2006). This definition refers to what is now regarded as tangible heritage and include paintings, sculptures, monuments, buildings and groups of buildings and sites containing works of man (Grandström, 2013).

Intangible heritage in the form of ―practices, representations, expressions, knowledge, skills instruments, objects, artefacts and cultural spaces associated with communities, groups and individuals" (UNESCO, 2003) have today been added to the term cultural heritage as they are seen as an integral aspect of heritage significance (Ahmad, 2006).

Additionally, the term digital heritage is used to define digital heritage material such as websites, movies, and games, and is provided for by UNESCO (UNESCO, 2003).

Ultimately, as expressed by the Venice Charter 1964, it is the responsibility of each country to define its cultural heritage in accordance with its traditions and culture and subsequently to preserve this heritage for current and future generations based on internationally agreed principles (Venice Charter, 1964).

Cultural heritage sites are the focus of this research and are defined by the ICOMOS Ename Charter as follows: ―Cultural Heritage site refers to a locality, natural landscape, settlement area, architectural complex, archaeological site, or standing structure that is recognised and often legally protected as a place of historical and cultural significance" (ICOMOS, 2007).

2.1.1 Digital documentation of cultural heritage sites

Cultural heritage sites, as described above, are important to individuals' and cultures' identity and sense of place. Unfortunately, many cultural heritage sites have been damaged and destroyed due to man-made and environmental events throughout history. Today many sites are badly neglected and are in danger of deterioration (Held, 2012). It is therefore of great importance to digitally record these sites for the benefit of future generations (Rüther et al., 2012).

The Zamani project, of which the author is a member, has been digitally documenting heritage sites since 2006 and has developed efficient methods of data capture and processing. The Zamani project is a not-for-profit research project, founded by Emeritus Professor Heinz Rüther, and based in the Geomatics Division at the University of Cape Town. It is currently funded by the Zamani Project Cultural Heritage Sites Trust (Rüther et al., 2012).

Techniques used for the digital preservation of heritage sites include: 3D laser scanning, photogrammetry, GPS and conventional surveys, remote sensing, and panorama photography. The acquired data is used to create textured 3D models, Geographic Information Systems (GIS), plans, sections and elevations, and 360-degree panoramas (Rüther et al., 2012).

Sites are digitally recorded for:

- education and research
- conservation and restoration
- site management and tourism
- and as a record for the future

2.1.2 Documentation of the World Heritage Site of Petra by the Zamani Project

Petra, in Jordan, is a UNESCO World Heritage Site located between the Red Sea and the Dead Sea. It rose to prominence in 300 BC as the capital city of the Nabataens, a nomadic tribe, who created their empire at a juncture of major trading routes at the time. The Petra Archaeological Park (PAP) consists of rock-hewn tombs, tricliniums, and dwellings as well as freestanding palaces, colonnades and temples. The monuments of Petra have, throughout their existence, been threatened by catastrophic events such as earthquakes and floods as well as slow natural weathering due to temperature variations, wind and rain (Rüther et al., 2014).

The Zamani project has been spatially documenting Petra as a UNESCO Partner in the SIQ stability project, a ―Funds in Trust” project of the Italian Ministry of Foreign Affairs for UNESCO. The project is managed by the UNESCO Amman Office and has as main partners Italian geological experts from ISPRA (Italian Institute for Environmental Protection and Research - Geological Survey of Italy), the Zamani Research Group at the University of Cape Town, and the Petra National Trust (PNT). It is undertaken in cooperation with the Department of Antiquities of Jordan (DOA) and the Petra Development and Tourism Region Authority (PDTRA) (Rüther et al., 2014).

The principal objectives of the project are the development of a:

- Monitoring system aimed at detecting potentially unstable rocks and at-risk areas.
- Guidelines for implementation of sustainable landslide mitigation strategies and for management of the Petra area.

- A GIS platform for storage, analysis and management of data relevant for the Petra Archaeological Park area.
- A 3D computer model of the Siq and the major structures and landscape of the site.
- A virtual world of Petra.

The Zamani group was responsible for “the 3D modelling of the Siq, the 1.2 km access route to the site, based on terrestrial laser scanning and aerial photography, the creation of a comprehensive site GIS and database, a virtual world, and 3D documentation of the important tombs and structures” in the central valleys of Wadi Musa and Wadi Farasa in the Petra National Park, using laser scanning and photogrammetry (Rüther et al., 2014).

To date 25 billion surface points have been captured in over 2200 individual laser scans. These scans have been registered together and cover a distance of 3km. 30 major structures, the 1.2km Siq and much of the terrain have been recorded in 1cm point interval resolution and 3D surface modelled. Most of the structures have also been textured to create more photorealistic representations of the monuments.

A comprehensive GIS was created of the site consisting of all data captured on site and data resulting from previous archaeological missions by international teams. Included in the GIS is a 1m grid DTM created of the PAP using 15 aerial images and approximately 50 RTK GPS points. Also included are 450 360-degree panoramic images captured of the site.

2.2 Virtual reality

Virtual reality (VR) environments are designed to create a feeling of presence and navigability in a real world that is, in fact, entirely simulated. The virtual environment (VE) can be a representation of the real world or an imagined world (Steuer, 1992; Aziz and Siang, 2012, Noh et al., 2009). VR is defined as “the use of a computer-generated 3D environment – called a virtual environment – that one can navigate and possibly interact with, resulting in real-time simulation of one or more of the users five senses” (Guttentag, 2010). This impression of reality is referred to by DeLeon and Berry as “immersion”, or the “suspension of disbelief” (DeLeon and Berry, 2000). To achieve the feeling of immersion, VR often employs advanced technologies, such as head-mounted displays (HMD), sound, and motion-sensing gloves to create a high-level of interactivity in virtual reality systems (Granström, 2013; Deleon and Berry, 2000). VR is a useful visualisation tool of places that are not physically reachable, no-longer exist, or need to be examined from diverse and unique points of view (Russou, 2002). Combined with the growth of computer technology and sophisticated 3D modelling and texturing software, VR is thus ideally suited for the visualisation of cultural heritage sites.

2.3 Augmented reality

Augmented reality (AR) is a combination of virtual reality and reality. The goal is to “create the sensation that virtual objects are present in the real world” (Cawood et al., 2007). AR allows users to see the real world with 2D and 3D virtual elements overlapping the real world display (Aziz and Siang, 2012).

Reality is either displayed on a screen and a virtual object overlaid on this, or the user perceives the virtual object overlaid on reality by looking at a mobile display or through an AR headset such as an HMD or glasses (Cawood et al., 2007).

AR technology can be used to reproduce on-site virtual reconstructions of monuments (that are registered to their correct locations in 3D space) enabling a user who is visiting the site to visualise what a monument could or would have looked like (Noh et al., 2009).

2.4 Virtual heritage

The term virtual heritage (VH) is derived from the terms virtual reality and cultural heritage (Granström, 2013). VH can be described as “the use of computer-based interactive technologies to record, preserve, or recreate artefacts, sites, and actors of historic, artistic, religious, and cultural significance and to deliver the results openly to a global audience in such a way as to provide formative educational experiences through electronic manipulations of time and space” (Stone and Ojika, 2000).

VH is then the integration of digital cultural heritage data into a VR application to create a simulated virtual heritage site environment (Roussou, 2002; Sanders, 2014). This definition of VH has, in the past, been used to refer to renderings, animations, panorama tours, and movies. However, without a fully 3D computer generated environment where a user is able to navigate (as defined in section 2.2), these elements are not considered virtual reality (Sanders, 2014).

VH can then be used as a platform for visitors and researchers to learn and understand about cultural heritage sites without having to physically visit these sites (Noh et al., 2009). Some examples of the uses of VH include excavation documentation, education, publication, museum display, serious games and tourism (Sanders, 2014).

The relations between VH and associated fields are described by the diagram below.

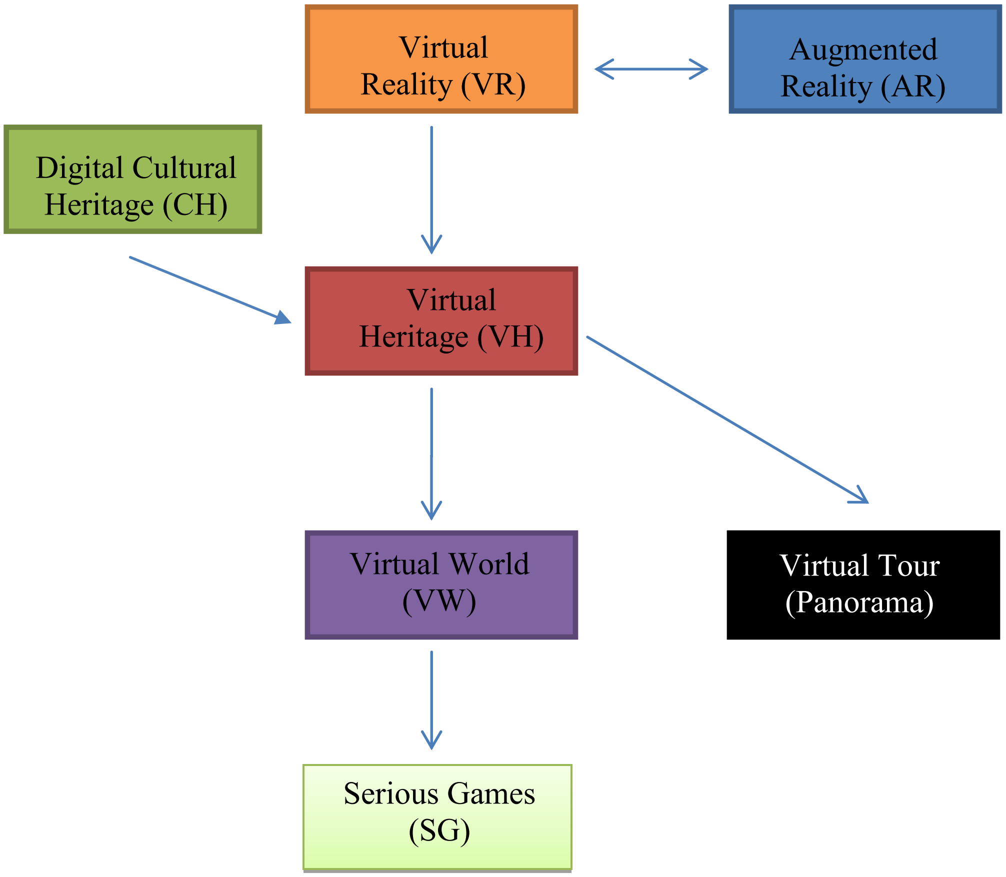


Figure 2.1: Diagram of the virtual heritage hierarchy.

2.4.1 Relevance of virtual heritage

Digital libraries have become prevalent in storing and displaying digital heritage data. Museums and libraries have collected vast quantities of digitised heritage data that can be accessed by the public and used as educational tools (Ott et al., 2010). Examples of this include the Musawwarat Graffiti Archive where users can view images of individual graffiti online (Musawwarat Graffiti Archive, 2011), The Trust for African Rock Art which hosts a large database of African Rock Art (TARA, 2014), and the Smithsonian X 3D archive where individual 3D models of heritage artefacts can be interactively viewed online (Smithsonian Institute, 2014).

Museums have come under pressure to make their collections more accessible to the public because of a need for better understanding of human history and heritage. Informal places of education such as museums, cultural centres, and the media can benefit greatly from virtual heritage which seeks to display digital heritage data and raise public awareness of heritage sites (Roussou, 2002).

VR has proven to be a valuable visualisation tool for spaces that cannot be easily accessed, sensitive environments that are not suitable for crowds, and for structures that need to be examined from different perspectives. They can thus act as an alternative to actual site visitations. In a VH system it is important to represent and display heritage data that is authentic and accurate, as this is what is expected by the public, in the same way as museums are expected to present data that is reliable (Roussou, 2002; Sanders, 2001; Aziz and Siang 2012).

Another example of VH is The Hellenic Cosmos Cultural Theatre which offers visitors a fully 3D immersive environment in a virtual reality theatre. Visitors wearing 3D stereo-scopic glasses are able to take part in virtual travels, visiting cities and monuments that no longer exist (Foundation of the Hellenic World, 2014). Champion (2008) argues that ―the primary aim of virtual heritage is to communicate the cultural significance of a site" and that this should be done via interaction (Champion, 2008).

To summarise, five target groups of users of VH content have been identified by Sanders (2001). These are: (1) archaeologists and historians who are interested in understanding the past, (2) museums and on-site interpretation centres, which the public visits to experience and learn from, (3) schools and education professionals, (4) experimenters who are individual testers or students, (5) the entertainment industry which includes computer games and the media (Sanders, 2001). To this another group can be added; conservators and restoration experts. Virtual heritage thus encompasses a wide audience and has a range of varied needs arising from different applications. One of these needs is the virtual world described in the following section.

2.5 Virtual world

For the purpose of this research the term virtual world (VW) will be used to describe a VH environment which is a fully immersive 3D virtual space with user interactively (Champion, 2008; Sanders, 2001). This is to distinguish from panorama tours, animations and videos which limit the user to a predefined path or tour, and are often referred to as virtual tours. The relation between virtual tours and virtual worlds can be seen in figure 2.1. A VW is a 3D VH application that uses virtual reality to simulate a 3D cultural heritage environment where a user is able to navigate freely. This is described in detail below.

2.5.1 Description of an ideal VW of heritage site

The following is a description of an ideal VW of Petra as imagined by the author. The ―visitor” is placed in the virtual environment of a 3D heritage site scene with a first-person perspective of the site‘s terrain, seeing the scene and its monuments as it would be seen in real life. The terrain should be of the highest detail possible with a high resolution texture. Using the keyboard and/or mouse, or other navigation device, the visitor should be able to manipulate the view to create a sense of place and the atmosphere of walking through the scene (flying could also be incorporated). The visitor should then be free to "visit" 3D heritage structures, which in Petra comprise of tombs, caverns and buildings. When arriving at a structure the visitor should be able to view 360-degree photographic panoramas of the environment to add a sense of reality. The visitor would then be able to explore the textured 3D model of the structure by entering and exploring at will. Information panels should be available giving descriptions of the structure and videos of the structure could also be made available (Wessels et al., 2014).

The visitor is free to explore the landscape of the virtual environment and the other structures within it. There should be minimum restrictions placed on where the user can explore as this would conflict with the feeling of being in a simulated real world. A top-view mini-map should be visible in a corner of the users‘ computer screen to indicate where the user is inside the virtual environment. The map should contain clickable objects allowing the user to move to new points of interest in the scene (Wessels et al., 2014).

The VH must consist of high quality data, be simple to navigate, and be interesting and informative to explore. Access to the VW should be as easy as possible and thus should be made freely available on a web-browser to allow an international audience to use the VW. (Wessels et al., 2014).

The above is designed as a baseline VW where all spatial data is incorporated and navigability and interactivity are available to explore the site. However, in an ideal VW additional functionality should be added depending on the target audience. This includes: video and audio narration, educational games, virtual tour guides, scientific analysis, multiple users, and other functionality as required. These aspects are beyond the scope of this research and it is hoped they can be added at a later stage (Wessels et al., 2014).

2.5.2 The need for a VW of Petra

The digital spatial materials generated by the Zamani project of Petra should be used by as wide an audience as possible to achieve the maximum benefit from the material produced. Petra can benefit from a VW which displays these materials in a manner that any user, inexperienced with viewing 3D data, can use. A user would be able to explore Petra virtually, before, after, or in place of, a site visit to gain a better understanding of the site. It is the objective of this research to create such a world.

Users of the VW specific to Petra would include: The general public who would explore the site from a touristic point of view; scholars who would use it as an educational tool; heritage professionals, and site management officials would use the VW to gain an understanding of the spatial aspects of the site, and visualise all the spatial data in a seamless environment to perform analysis and make managerial decisions. The VW could also be used to help with funding proposals.

2.6 Problem statement

Spatial data of heritage sites are currently not being exploited to their full potential. The reasons being that many operators/users are not familiar with manipulating spatial data, especially 3D data, and that current viewing platforms are not able to accommodate the many spatial data types simultaneously in an easy to use manner. e.g. 3D models, GIS data and panorama imagery must each be viewed in separate viewing platforms, making them individually less effective. Additionally spatial data is currently not presented in a form that is appealing to the general public.

2.7 Primary objectives

This research's primary objective is to design and create a VW of the site of Petra, Jordan, incorporating spatial data gathered on site and processed by the Zamani team. This captured data consists of 3D laser-scans, differential GPS points, photogrammetric and panoramic photography.

The VW is to comprise of the following spatial data that were produced of the site from the data captured on site: A DTM and orthophoto, textured 3D models, 360-degree panoramas, individual photos, maps and plans, and any other spatial and attribute information that is available.

One of the challenges of creating the VW will be the amount of data that can be incorporated due to computer hardware limitations. 3D models, consisting of millions of triangles will have to be intelligently decimated and visualised to allow for a good compromise between fast computer running speeds and good visual appearance.

The VW should be available as a stand-alone downloadable application, accessible in an online web-browser, displayable in a visitors centre or classroom, and potentially be available on a mobile device. The VW should be interesting and enjoyable to the general public.

2.8 Research questions

The following research questions can be derived from these objectives:

- How to integrate 3D spatial data that is available of heritage sites into a seamless virtual environment?
- How to create 3D spatial content with photorealistic textures for a virtual environment of a heritage site?
- How to integrate 360 degree panoramas into a 3D environment?
- How to create a VW that has ease of use. i.e. easy to navigate, easy to use its functionality and easily accessible?
- How to make the VW an engaging and immersive experience where a user would be eager to explore the virtual heritage site?

2.9 Limitations and phases of the VW creation

The limitations of the VW need to be set by the author for this research as there is a limitless scope of what can potentially be created. The many different target audiences, each with their own needs, make it difficult to create a single VW to cater for all. Additionally PC hardware capabilities need to be determined as well as level of programming skill ascertained so that the expectations of the end product of the VW are managed correctly.

The first limitation is that the software used to create the VW must be free and easy to use for a non-professional programmer.

The second limitation is that the VW must be able to be accessed online, or downloaded onto a standard PC and be able to run without installing any software. No expensive computer hardware must be required to run the VW. This will pose challenges in the amounts of detail that can be displayed due to graphics card and memory limitations.

The third limitation is to create a VW that functions as a baseline virtual environment of Petra where a user can freely navigate around in an easy manner and view all spatial data that is available. The VW will seek to create an environment that is as real as possible for a user to explore as if they were walking around the site in person. At this stage the VW will have no multi-player capabilities, no educational game aspects, nor virtual tour guides, and will only consist of spatial data of Petra produced by the Zamani project. The scope of this research ends at this stage.

2.10 Outline of the work

The work will begin with a literature review of the historical and current approaches to creating VH applications. Then a review of six VH applications which were created for other heritage sites will be undertaken. Following this an identification of input data and design elements typically used to create a VW will be carried out. The implementation of these elements to create the VW of Petra and a discussion of the VW will finally follow highlighting its strengths and weaknesses. Recommendations will then be given to improve the VW.

Chapter 3

Literature and software review

It has been predicted that in the future it won‘t be necessary to travel to remote historical places to experience them as we will be able to choose from a list of sites to enter virtually and visit from anywhere in the world (DeLeon and Berry, 2000).

There are many types of virtual heritage applications available today. The type and implementation of the applications depends on the target audience and the objectives of the individual project. Different varieties of VH applications will be explored in this chapter using significant case studies as benchmarks for progress in the VH scene. VH applications have been criticised in the past for their low levels of interactivity, and their lack of intangible heritage content, such as songs and ceremonies. Thus VH practitioners have started looking towards software used to create video games for solutions to address these problems as video games offer more advanced virtual recreations incorporating intangible content, and greater interactivity, while being able to run on fairly low performance machines (Granström, 2013). Video games and VH have many technical similarities, however, there is a difference between them which is that VH has been confined to research projects and industry labs due to lack of funding and narrow research objectives (Roussou, 2002), while video games target the general public. Many of the case studies listed below use game engines in their development.

3.1 What makes an effective VH application

To define what makes an effective VH application is a subjective exercise based on the user‘s unique expectations and experience of the application. However, there are basic elements that should be present in a VH application to ensure its effectiveness. El-hakim et al. (2004) determines that the following eight requirements should be satisfied; ―high geometric accuracy, capture of all details, photorealism, high automation level, low cost, portability, application flexibility, and model size efficiency”.

Gramström (2013), in the study ―Elements for Games in VH applications” conducted an analysis of journal articles regarding VH and subsequently identified elements in computer games that are seen as important to VH applications. These elements should be kept in mind when developing VH applications to satisfy the general target audience in the same way that video games cater for their demands. The elements are divided into five categories.

3.1.1 Interactivity

Interactivity refers to the amount of control the user has in influencing the virtual environment. It is achieved by ―the ability to affect, use or communicate with something or someone in a digital or virtual environment” (Gramström, 2013). This can be accomplished through exploration, tasks, dialogue or quiz provided in the virtual heritage application.

3.1.2 Meaning

The field of VH has been criticised for focusing too much on tangible heritage. Equally important is intangible heritage which adds meaning to VH applications. This can be achieved by educational materials on the culture and history of a heritage site, and can be conveyed through a story.

3.1.3 Player Character

VH applications can offer the experience of being a relevant character in a historical virtual environment. In this way the user is able to learn more about living in a historical time by experiencing first-hand how a historical figure would have lived their life by actively participating in the story being told. The user would be given an avatar that represents their character.

3.1.4 Other characters

Cultural presence can be enhanced by introducing other historical figures that the user can interact with. It is also possible to introduce multiple players into the virtual environment where multiple people are able to be present in the same virtual environment.

3.1.5 Accuracy and Realism

Authenticity is of great importance in cultural heritage. As virtual heritage application only allow for recreations of actual cultural heritage sites it is therefore impossible to be completely authentic in a VH application. The basis for accuracy and realism of a VH application is cultural and historical accuracy and realism. The environment, buildings, objects and characters should all be as accurate and real as possible to make the VH application authentic. The cultural and historical aspects of the virtual heritage application can be regarded as cultural presence. Thus the visual elements of a reconstructed heritage site, the behavioural aspects of the characters present, the audio and smell of the site (where applicable) should all be as accurate and real as possible to create an authentic virtual heritage site experience.

3.2 Panorama tours

Currently, the most widely employed method to view heritage sites virtually is by means of a panorama tour (See figure 3.1). A panorama tour, often, and somewhat misleadingly, referred to as a ―virtual tour”, typically displays a map layout of the site and next to this a window showing the panorama where the user can pan through 360 degrees from one viewpoint. The map layout displays points marked where panoramas are situated and the user can ―jump” between these positions and view the site from these spots. Although this is a suitable tool for a general impression of the site and its environment, panorama tours lack full freedom to navigate the site as the user is confined to specific viewpoints. More importantly, the panorama tour does not immerse the user into a true 3D environment (Wessels et al., 2004).

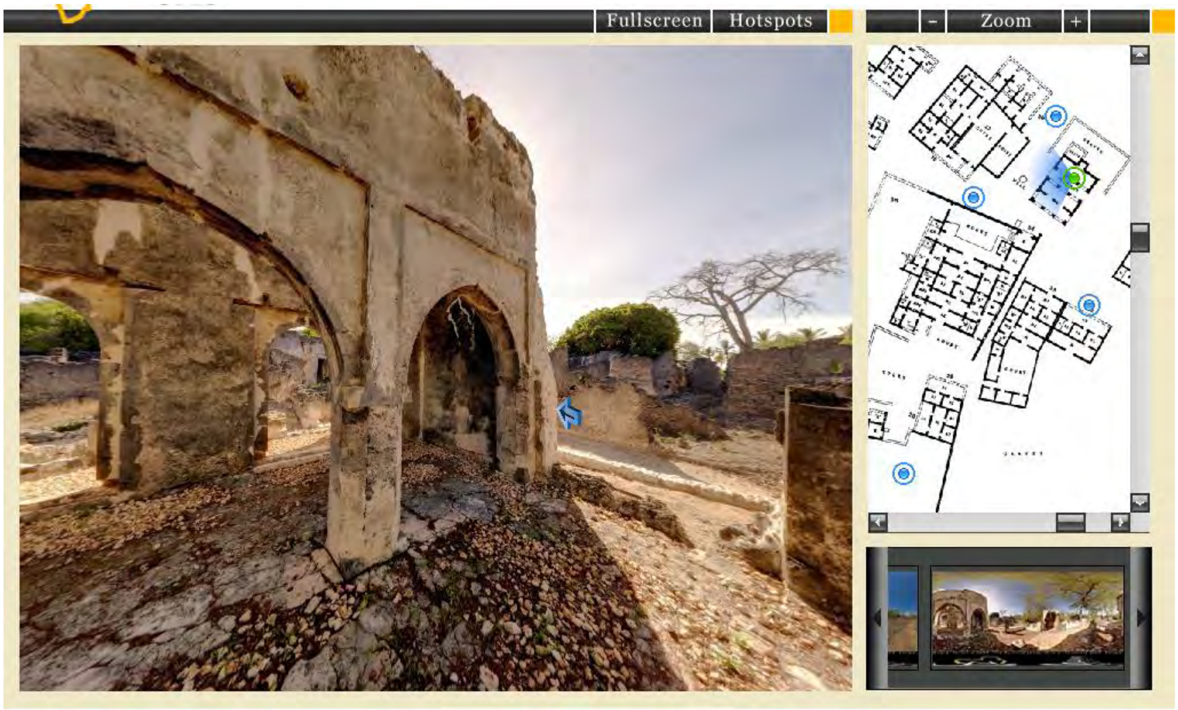


Figure 3.1: An example of a panorama tour.

3.3 The major institutions in the VH world

Many VH institutions have emerged recently due to computer hardware and software advancements and the public interest in VH. Sanders (2013) lists www.publicvr.org, www.learningsites.com, www.vizin.org, www.virtualheritage.net, the International Virtual Heritage School; and the 1st International Congress on Digital Heritage, Marseilles, 2013, as examples of such institutions (Sanders, 2014). The first two listed institutions are discussed below.

3.3.1 PublicVR

PublicVR conducts research in virtual reality for education and is headed by Dr. Jeffrey Jacobson. Virtual worlds are hosted on the PublicVR website which can be run on a web-browser or downloaded. These include: Virtual Theatre district of Pompeii project, Virtual Egyptian temple, and the Vari house VH heritage applications (in collaboration with

LearningSites, discussed below). PublicVR are also involved in virtual reality installations such as the EarthTheatre DigitalDome, a panoramic display at the Carnegie Museum of Natural History, Pittsburgh, Pennsylvania, USA which seats 60 people within a 210-degree arc of its curved screen and plays virtual guided tours of heritage sites. PublicVR exists online only, with no offices, and projects are done on a contract basis or in collaboration with other institutions. The projects are funded by research grants and donations. Support is offered to collaborators to help find grants, provide legal framework and distribute the end products (Publicvr, 2008).

3.3.2 Learning Sites

―Learning Sites”, led by Dr. Donald Sanders, has been in existence since 1997 and regards itself as ―the Leader in Reliable Archaeological Visualizations in Interactive Education and Research” (Learningsites, 2014). It grew out of the process of digitally preserving records, such as old photographs and diagrams, of heritage sites and then displaying this digital data as educational material in the form of multiplayer, multimedia experiences in museums and on-line. Their goal is to virtually reconstruct heritage structures by creating interactive 3D models based on actual archaeological evidence. The work is used for teaching, research, archaeological fieldwork, museum exhibitions, and tourism. Their clients are museums and universities, and besides web-based VWs, museum display kiosks and HMDs are utilised. Projects include: Northwest Palace of Ashur-nasir-pal II, Nimrud, Assyria (1999); Gebel Barkal Temple B300, Sudan; Seyitömer Höyük, Kütahya, Turkey. The VWs that the author was able to access appeared outdated and of poor quality by today‘s standards, with the Vari house being an exception.

3.4 Methods of creating virtual heritage environments

There are many software packages for creating and displaying a VW on a desktop computer, either with a standalone, downloadable application or online. The most popular software is VRML/X3D, Unity and The Unreal Engine. Additionally O3D, OpenSpace3D, and TurnTool are platforms for bringing 3D content online (Koutsoudis, 2012).

3.4.1 VRML and X3D

Virtual Reality Modeling Language (VRML) was first developed in 1994 as the international standard for describing and displaying virtual reality 3D scenes and worlds on the internet (Champion, 2014). The VRML standard has had many updates since its release. Extensible 3D (X3D), an extension to VRML, has evolved and superseded VRML and is managed by Web3D (Web3D Consortium, 2012). Both VRML and X3D have been accepted by the International Organisation of Standards (ISO) (Koutsoudis, 2012).

VRML environments can run on web-browsers once a plug-in is installed. Early VH applications, such as Gebel Barkal Temple B300 (learning sites), were created using VRML. However, they are criticised for being slow and buggy (Champion, 2014). X3D is intended to work with the latest web coding language XML5 that brings 3D content to the web without the need for plugins (Koutsoudis, 2012).

3.4.2 Game engines

A game engine is a software framework designed for the creation and development of video games for consoles, PC‘s and mobile devices. They typically offer a rendering engine (displaying 3D data on a screen), physics engine (calculations of real world physics in a virtual environment, such as gravity, that effect virtual objects), sound, scripting, animation, AI, networking, and other functionality as would be used in the creation of a video game. They provide a set of development tools in an integrated development environment (IDE) (source code editor). Game engines are sometimes referred to as ―middleware” as they provide a re-usable, flexible, software platform for developers to create a game without the need to create engines from scratch for each game (Svånå, 2010).

Game engines thus offer the VH community great opportunities in creating interactive and engaging virtual environments of heritage sites without having to create software from scratch. As DeLeon and Berry (2000) note ―video game companies have made great strides in this area (of presenting 3D virtual environments). They have spent millions of dollars developing real-time 3D engines that focus on a series of new core technologies specifically dealing with presenting complex 3D environments, comprised of textured and shaded polygon-based worlds, to a low-end audience running standard personal computers".

Game engines are highly suitable for VH applications due to their ease of modification and addition of functionality to virtual environments using scripting, high visual appearance owing to their lighting, physics engine and animation systems, large user communities, networking abilities, and virtual scene creators (Svånå, 2010; Jacobson, 2005a).

Game engines also allow for the introduction of virtual characters into virtual environments in the form of a bot (short for robot). A bot is an avatar that possesses weak AI which controls the avatars actions and movements in the environment and simulates human-like intelligence (Patel, 2012).

Dozens of game engines are available to the VH community today, with each game engine offering a unique solution to VW creation. The two most popular game engines in use in VH applications are the Unreal Engine and Unity 3D (Svånå, 2010). Both are free and relatively easy to use for non-professional programmers.

3.4.2.1 Unreal Engine

The Unreal Engine was developed by Epic Games with the first game, Unreal, released in 1998. The engine was developed mainly for first person shooter (FPS) games. The first version of Unreal Engine incorporated rendering, collision detection, bot AI, visibility, networking, scripting (UnrealScript) and file system management. UnrealEd is a level editor (framework to create virtual environments) also included with Unreal Engine (Unreal engine, 2015).

In 2009 Epic released the Unreal Engine 3 Software Development Kit, which it called the Unreal Development Kit (UDK), available for free to create non-commercial games and applications. The latest version is Unreal Engine 4, released in 2012, which can operate on desktop computers, gaming consoles, smart phones and Web browsers (Unreal engine, 2015). There is a large community of developers who produce freely available content to be used in the Unreal engine such as virtual environments, people and objects (Jacobson and Holden, 2005).

3.4.2.2 Unity

Unity is a ―game development ecosystem" (Unity3D, 2014) commonly used to create games for web-plugins, desktop platforms, consoles and mobile devices. It is a cross-platform engine (meaning the output can be used on iOS, Mac, Android, Apple and others) with its own Integrated Development Environment (IDE) (Brodkin, 2013). There are two versions of the Software; Unity which is free for non-commercial use, and Unity Pro which has advanced features, such as advanced lighting and level of detail capabilities. All programming is done via scripts which are attached to objects in the game. The scripts are written either in Java or C++ (Unity 3D, 2014). There is an online forum where developers using Unity can pose questions to other users and to Unity creators. There is a good user manual as well as many tutorials. The Unity community has 2.5 million registered developers and 500 000 active developers as of March 25th 2014, which would indicate that it is well accepted in the virtual reality community (Unity 3D, 2014). Unity provides a plug-in to run projects on web browsers and is also able to create downloadable stand-alone applications (Koutsoudis, 2012).

3.4.2.3 Comparison of Unity and Unreal Engine

	Category	Unity	Unreal Engine
1	Price	Non-pro version free	$19 a month
2	Coding languages	C#, JavaScript	C++, UnrealScript
3	Desktop and mobile publishing platforms	Windows, Mac, Android, Windows phone, iOS, Linux, Webplayer, Xbox, Oculus rift and more	Windows, Mac, Android, iOS, Linux, Oculus rift and more
4	Webplayer	Unity Web Player plugin enables 3D content to run in a browser	No web player plugin
5	Graphics	Online reviews quotes Unreal as having better graphics than Unity	
6	Learning curve and ease of use	Online reviews quotes Unity is easier to learn and use than Unreal	
7	User support	Big community of users. Good support from Unity. Large forum	Good and fast support from Unreal Engine team.

Table 1: Comparison between Unity and Unreal Engine. Information as of April 2014, from (Unity 3D 2014; Unreal Engine, 2015; Mayden, 2015; Unity vs Unreal Engine 4, 2015; Bailey, 2015; O' Flanagan, 2015).

3.5 Serious games

Serious games (SGs) are described as –games whose primary goal is not entertainment” (Bellotti, 2012) but education and training (Foni et al., 2010). Games (video, console, PC) –offer an exploratory environment in which learners can engage in active problem solving” (Lucey-Roper, 2006). Students learn and remember things much more effectively by performing a task rather than by reading about it (Bellotti et al., 2012) and so a SG offers an effective method of learning by creating motivation to learn (Lucey-Roper, 2006; Champion, 2014). Belloti et al (2012) have developed the Sand-Box Serious Games model which –invite players to perform cognitive tasks contextually, while exploring information-rich virtual environments” (Bellotti et al., 2012, pg2). SGs are thus attracting more interest with educators as they use the latest simulation and visualisation technologies as well as exploit the fact that a growing number of people today play video games (Bellotti et al., 2012; Champion, 2014). –They allow players to participate in new roles and contain intrinsic motivation through fantasy, challenge, curiosity and competition” (Lucey-Roper, 2006). Thus serious games based on VH data can combine to create a stimulating learning environment for the technology driven youth of today. Figure 2.1 shows how serious games fit into the field of VH.

3.6 Methods of viewing VH applications

There are a variety of methods that can be used to view VH applications. Some of these are discussed below.

3.6.1 Window on the world

Window on the world (Wow) applications display a virtual world on a computer monitor screen with the user typically navigating through the environment by way of a joystick or keyboard and mouse combination as can be seen in figure 3.2. The environment is seen as if the user is looking at a scene through a window (Pujol, 2004). This is the most common type of method to view VH applications with the environment being displayed in either first or third person perspective. The level of immersion using this method is low (Pujol, 2004).

Figure 3.2: Window of the world method of viewing a virtual world on a computer screen. From (Pujol, 2004).

3.6.2 Head mounted displays

A Head mounted display (HMD) is a display device worn on the user's head that displays an image in front of one, or each eye. This can be seen in figure 3.3. HMDs visually immerse the user in a virtual environment scene by cutting out all real world views and only displaying the virtual environment. A HMD is often equipped with a system that tracks the angle and orientation of the HMD so that as a user moves their head the view in the virtual environment changes as it would in real life.

Figure 3.3: Head mounted display device. From (Pujol, 2004).

3.6.3 Virtual theatres and digital domes

Virtual theatres are large screen displays of virtual environments. The screens are typically curved to create a feeling of immersion in a virtual environment as can be seen in figure 3.4 below. A digital dome is similar to a virtual theatre except the screen is a spherical or partly spherical display that projects visuals of a virtual environment onto its interior surface. Virtual theatres and digital domes are intended for groups of large audiences where the screen fills most of the users field of view, visually immersing them in the virtual environment in a scale of 1:1 (Pujol, 2004; Jacobson and Holden, 2005; Jacobson, 2013).

Figure 3.4: A Virtual Theatre. From (Jacobson, 2013).

3.6.4 Cave automatic virtual environment (CAVE)

A cave is most commonly intended for use by a single person and consists of multiple flat screen displays arranged in a cube that encloses the user and displays a virtual environment (Jacobson, 2013). An example can be seen in figure 3.5 below. A user will often wear 3D projection goggles inside the CAVE to see a 3D display of the envrionemt. Users can then see ‗floating' objects in the CAVE that they can walk around. The users movements inside the CAVE are continuously tracked and the display is continuosly adjusted to retain the users perspective of the scene (Cruz-Neira, 1993).

Figure 3.5: Image of a user inside a CAVE. From (Wikipedia, 2015e).

3.7 Case studies

The following six case studies of VH applications are presented in order of the projects release dates.

3.7.1 Dudley castle virtual tour

3.7.1.1 Background

The first example of a virtual tour in a museum was the Dudley Castle virtual tour, a computer visualisation of the castle as it appeared in 1550 (Johnson, 2005). The term ―Virtual Tour” was coined by Johnson who combined the term virtual reality and royal tour. It was opened by Queen Elizabeth II in 1994 and ran until 2005 in the visitors centre in the remains of the Dudley Castle in England (Johnson, 2005).

3.7.1.2 Description

The tour consists of computer rendered still images of reconstructed scenes based on photographs of today‘s ruins. A user can click through the images to navigate through the ruins as they might have appeared in 1550. There is no 3D navigation capability in the tour, only the ability to view static images.

Figure 3.6: Left: Image of what a visitor will see today. Right: Image showing a computer rendering of a reconstructed visualisation of the Castle. From (Johnson, 2005).

3.7.1.3 Useful design and data components

Being the self-proclaimed first virtual tour, this tour is understandably simplistic compared to recent advancements in the field of VH. There is no 3D walkthrough functionality, only the ability to view static scenes. It does however demonstrate the potential of 3D computer visualisations (by way of static renderings of the models) consisting of textured 3D models, of digitally reconstructed heritage sites.

3.7.2 Virtual Notre Dame Cathedral

3.7.2.1 Background

The VRND (Virtual Notre Dame) project, created in 1999, was one of the first virtual heritage applications to make use of a game engine (Svånå, 2010). It was created by Digitalo studios and Virtual Systems Lab in Gifu, Japan. The collaborators first created a test project, the Virtual Florida Everglades, to see what was possible with a standard desktop computer using a game engine. Epic Games‘ Unreal Engine was chosen for the production of the VW because it was deemed cost-effective and far advanced compared to other development environments. The Everglades project was a large environment populated with AI bot characters and displayed as a freestanding museum exhibit. The VRND project was then based on the experiences of using the Unreal Engine in the Everglades project. The developers wanted to create a reconstruction of the Notre Dame Cathedral and subsequently a fully immersive 3D virtual environment described by DeLeon and Berry (2000) as: ―no tricks this time - virtual meant virtual, not a series of stitched images or panoramics, but an actual 3D model with more ―oomph" than virtual reality Modeling Language (VRML) could deliver".

3.7.2.2 Description

The VRND project first reconstructed the cathedral in 3D manually, using as little amount of polygons as necessary for a realistic visualisation (without using point clouds created through laser scanning or photogrammetry) in a variety of 3D software packages and using textures to compensate for low polygon details (DeLeon and Berry, 2000). The model was then imported into the Unreal Engine along with AI characters. Users are then able to download a self-contained viewer, which is free and globally accessible, to run the VRND where they are free to explore the cathedral, take a guided tour with a virtual tour guide, learn about the history and heritage of the cathedral and take snapshots of scenes that are then saved onto the user‘s PC. Additionally, the user may log onto a server where VRND is housed and run the project in multi-player mode where the user can see and chat to other players in the virtual environment. The virtual tour guide present in the project is a fully animated character with simple AI which is done by scripting behaviours into the guide‘s character (DeLeon and Berry, 2000).

Figure 3.7: 3D Model of the Notre Dame Cathedral. From (DeLeon and Berry, 2000).

3.7.2.3 Useful Design and data components

Considering the time of its release, 2000, this VW has an vast amount of functionality. This includes:

- Walk and fly mode navigability
- Demo tour (according to Deleon and Berry, 2000), the author never had access to this functionality
- Virtual tour guide avatar providing the history of the cathedral
- Torch for dark areas
- Lighting effects, which creates presence
- Multiplayer mode where users can see and chat to other users
- Virtual objects (religious crosses)
- Ability to take and save photo-snapshots of the users view in the cathedral

3.7.3 Discover Babylon

3.7.3.1 Background

Discover Babylon is an educational video game created by The Federation of American Sciences (FAS), UCLA‘s Cuneiform Digital Library Initiative (CDLI) and the Walters Art Museum (WAM) (Lucey-Roper, 2006). According to the Federation of American Scientists (FAS) ―Discover Babylon TM uses sophisticated video gaming strategies and realistic digital environments to engage the learner in challenges and mysteries that can only be solved through developing an understanding of Mesopotamian society, business practices, and trade" (FAS, 2008). The game was developed by Escape Hatch Entertainment and uses Vicious Cycles commercial game engine. The game can be downloaded from www.discoverbabylon.org for free and loaded onto a user‘s computer. It features historically accurate virtual worlds and teaches learners about the cultural legacy of Mesopotamia through a fully immersive environment. The aim is to create an educational game for ages 8-14 that ―rivals commercial games in the quality of its graphics, storyline, pacing and character animations" (Lucey-Roper, 2006; Petty, 2007).

3.7.3.2 Description

There are two versions of the game; a short ―Kiosk" version museum setups, and an extended ―long game" version for home computers. The Long version of the game will be discussed here.

The Society of Biblical Literature website introduces the game as follows. ―Summoned to the museum under mysterious circumstances, you are sent on an urgent mission to pursue a time-traveling archaeologist through Mesopotamian history. Armed with a translation device and a PDA (personal digital assistant) that adjusts to your level of interest, you leap into the persona of Taribi, a 10 year old scribal student. It's 2300 BCE. You are in the city of Uruk . . . and you are late for school" (Petty, 2007).

To play the game the user controls a third-person player, who walks just ahead of the camera view, to walk around a 3D virtual museum. The storyline, described above, dictates that the user must fulfil certain challenges, such as finding objects hidden throughout the virtual museum, to get through the first level of the game. Once the challenges have been accomplished the user then explores and completes challenges in the virtual cities of Uruk in 3100 BC, Ur in 2100 BC and Kalhu/Nimrud in 870 BC. As the game progresses the user learns about life in Mesopotamia in a fun and engaging manner (Lucey-Roper, 2006). –The game features historically accurate virtual worlds; digital representations of museum and library objects, challenges to keep learners motivated and engaged, a question and answer tool that allows players to receive timely and relevant answers, and immediate assessments that let learners know they have mastered skills and content" (Lucey-Roper, 2006).

Figure 3.8: Left: Player in the Museum in Discover Babylon. Right: Scene from Uruk. From (Lucey-Roper, 2006).

3.7.3.3 Useful design and data components

The game was found to be fun and engaging to play. Taking on, indirectly, the identity of the player, by controlling an avatar, immerses the user into the game and gives a sense of what it was like to live a day in the life in these VWs. The storyline has been well thought up, and even though it is aimed at 8 – 14 year olds it was challenging enough to hold the authors attention. The audio was realistic and all the characters in the game personalised. The virtual scene is fairly low in detail, created from primitive geometry and draped with sample texture. Objects such as trees and people are basic representations. Useful design elements include:

- Captivating storey-line
- Avatar bot AI characters
- Different scenes/levels that a player must progress to
- Educational content
- Interesting audio narration
- Flora
- Realistic looking sky

3.7.4 Egypt Virtual Temple and Gates of Horus

3.7.4.1 Background

A publicVR project, the Virtual Egyptian Temple is a virtual reconstruction of a non-specific Egyptian temple using elements from a New Kingdom temple (Jacobson and Holden, 2005). The project is continually being developed and all work is free to the public for use (PublicVR, 2008). Early versions used VRML and then the Unreal Engine to create virtual tours of the temple. They were available for desktop computers and the Unreal Engine version could also be played in the Virtual Theatre, a panoramic projection-based display (Jacobson and Holden, 2005). The temple was designed as a tool for education about ancient Egypt.

The virtual Egyptian temple gives rise to the game ―Gates of Horus” aimed at ages 11-13. A 2005 version of the game used the Unreal game engine. A later version uses the Unity engine and has additional features such as virtual objects and ambient music (Svånå, 2010).

3.7.4.2 Description

In Gates of Horus the user is a young priest who is schooled by a virtual high priest about ancient Egyptian custom. The user is free to walk around the temple in a ―first person” view, where the view is from the eyes of the player. Certain objects, which have a golden glow to them, are clickable and provide the user with an audio narrative by the high priest about a specific historical custom or ritual. To advance in the game a multiple choice questionnaire must be answered which allows the user further access into the temple if due knowledge is demonstrated.

The game is also playable at the Earth Theatre of the Carnegie Museum of Natural History in Pittsburgh where a life-sized presentation is displayed on a partial digital dome (Jacobson et al., 2009).

Figure 3.9: Left: Gates of Hours courtyard. Right: Screenshot of Gates of Horus Game. From (Jacobson et al., 2009).

3.7.4.3 Useful design and data components

The following elements were found to be well executed:

- Convincing models with high resolution textures
- Historical audio narrative
- Background music provides a sense of presence in a spiritual place
- Glowing features in the game encourage a user to click on objects

3.7.5 Rome Reborn

3.7.5.1 Background

Rome Reborn is an international initiative to create 3D computer models of the entire city of Rome as it stood in 320AD. Models dating forward and back in time from 320AD will also be created to cover the development of Rome over the time period 1000BC to 550AD (Frischerconsulting, 2014). The project has been running since 1997 with collaborators consisting of the Virtual World Heritage Laboratory of the University of Virginia (VWHL), the UCLA Experiential Technology Center (ETC), the Reverse Engineering Lab at the Politecnico di Milano, the Ausonius Institute of the CNRS and the University of Bordeaux-3, and the University of Caen. The project is currently managed by Frischer Consulting, whose mission ―is to apply 3D technologies to the study and dissemination of cultural heritage throughout the world" (Frischerconsulting, 2014).

3.7.5.2 Description

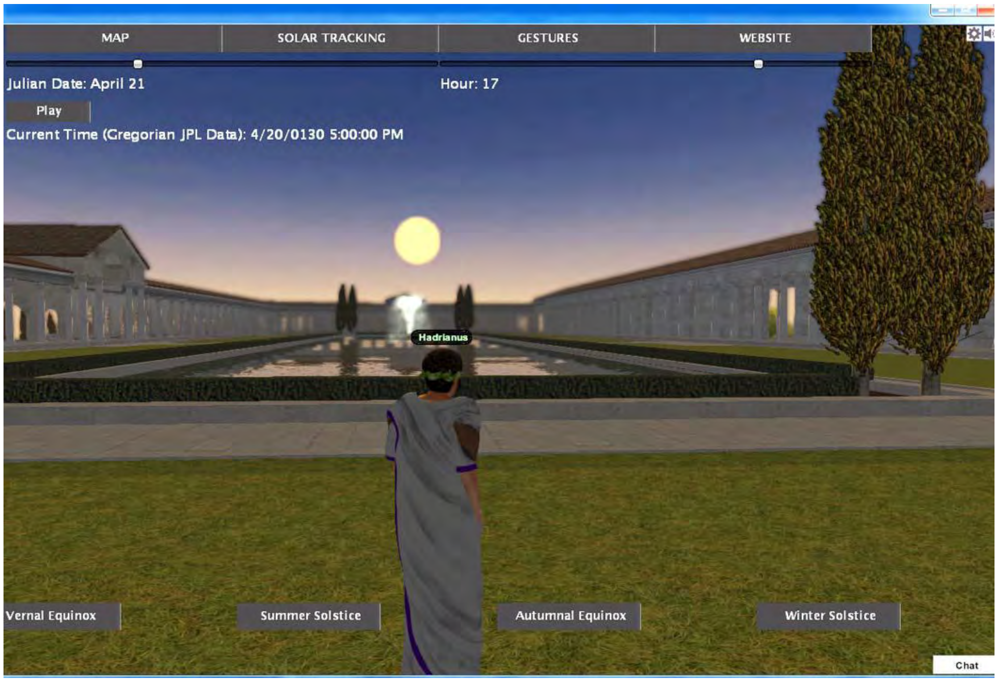
50 buildings and monuments have been modelled to date with the help of scientific advisory committees of experts. Other buildings are portrayed as simplistic structures as place-marks for future detailed models (Frischerconsulting, 2014; Dylla et al., 2008).

The model of ancient Rome was first shown publically in 2007 by displaying static images and video fly-throughs online. In 2008 Rome Reborn 1.0 was published in Google Earth as the layer ―Ancient Rome 3D". This layer has subsequently been removed. Recently methods of interactively presenting the model have been explored using a game-engine. A test project ―Hadrians Villa" was thus produced and it is now intended to apply the same methodology to the rest of the Rome Reborn project. A video fly-through of the latest version Rome Reborn 2.2 shows the virtually reconstructed city (Frischerconsulting, 2014).

The Digital Hadrian‘s Villa Project is a virtual reconstruction of the World Heritage Site of Hadrians Villa which lies 30km east of Rome. The VWHL used 3DSMAX to create a virtual restoration of the entire site. The Ball State University‘s Institute for Digital Intermedia Arts (IDIA Lab) then imported the model into Unity so that it can be interactively explored online in a VW. To attain authenticity of the VW the goal of the project was to ―ensure that all the main elements - from terrain, gardens, and buildings to furnishings and avatars - were evidence-based" (Frischer and Fillwalk, 2012).

Using this virtual world an archaeoastronomical study was made on the hypothesis that certain parts of the Villa were aligned to the suns apparent path on significant dates such as the solstices.

Frischer explains —We therefore programmed a plug-in for Unity3D that controls the movement of the sun on the sky dome in a way that accurately reflects the azimuthal data for the sun in the year 130 CE as seen at the precise geographical coordinates of Roccabruna. The solar tracker, or virtual heliodon, that we created as a response to this research, was envisioned as a simulation that would be a bridge between the virtual environment and coordinates from an external database calculating solar positions" (Frischer and Fillwalk, 2012). The results confirmed the hypothesis and in the online VW a user is able to change the date of the year as well as the time of day to show the position of the sun at that time. Certain alignments can then be seen as in figure 3.10 below.



Julian Date: April 21

Hour: 17

Current Time (Gregorian JPL Data): 4/20/0130 5:00:00 PM

Figure 3.10: "View of the Pecile, a large colonnaded garden. The imaginary line of the main axis of the built space runs from the back of the avatar in the foreground to the jet d‘eau in the background. At 5:00 pm, the sun aligns with the axis of the Pecile on April 21, 130 CE (Julian calendar)" (Frischer and Fillwalk, 2012).

Multiple users can access the VW online at the same time where they can interact via chat. The user is able to choose between seven avatars and their own player name. Non-player avatars with bot AI also roam the virtual world.

Figure 3.11: Left: Courtyard of Piazza d'Oro, Hadriaans villa today. Middle: Virtual reconstruction of Hadriaans villa. Right: Screenshot of the Hadriaans villa VW. From (Frischer, 2012).

3.7.5.3 Useful design and data components

- Avatars such as slaves, members of the imperial court and guards walk around the VW
- Multiplayer mode where avatars interact via chat
- Background sounds add to the feeling of presence.
- Interactivity provided by the sky-dome where a user can program the position of the sun

3.7.6 The Virtual Pompeii Project

3.7.6.1 Background

The ―Virtual Pompeii” project was initiated in 1995 with the aim to virtually reconstruct the Theatre District of the ancient Roman city of Pompeii. This was achieved at the Studio for Creative Inquiry at Carnegie Mellon. The reconstruction included 3D models of the Grand Theatre, the Temple of Isis, the Triangular Forum and connecting areas, known together as the

―theatre district", as well as historical supporting documents, and dramatic music. It produced a historical interactive recreation using immersive virtual reality technology. The output of the initial project was made freely available to the public via the web, or downloadable, for educational and artistic projects. This was in the form of a textured VRML model that could be run on standard computers and which allows for virtual navigation. The VW can be used as an education tool to show learners what the site looked like in the past. It was proposed that the model could be used for the creation of: an educational game, a virtual meeting place where people could log in and interact, or for museum-based education (Jacobson and Vadnal, 2005).

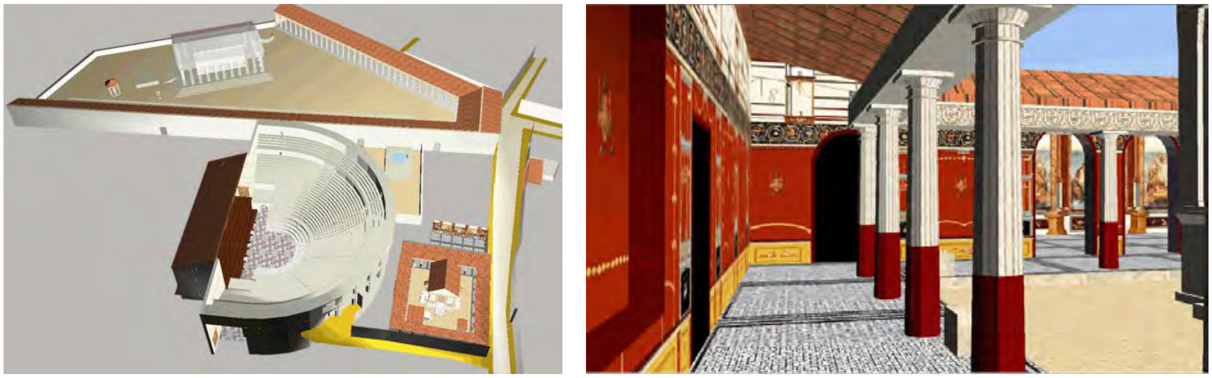


Figure 3.12: Left: Theatre District in the Virtual Pompeii reconstruction. Right: Virtual Temple of Isis. From (Jacobson and Vadnal, 2005).

It was later decided that VRML ―lacked the expressive power needed for accurate reconstruction" (Weiss et al., 2010) and so in 2005 a new project was initiated using Unreal Engine and Unity to construct a new model of the theatre district using new photographs and measurements of the site. The objectives were defined by Weiss: ―This new model is web based and it will be the centrepiece of an innovative web site that allows the cyber visitor to compare individual elements from the model with photographs and historical images. Eventually, the user will be able to navigate efficiently between the virtual space, the data, the metadata, and the way these data have been interpreted" (Weiss et al., 2010).

3.7.6.2 Description

The author could not find evidence of this web based comparison as described here. It is stated that the project is on hold for now (PublicVR, 2008). What is available is a virtual world of the Theatre District of Pompeii created in Unity and using 3D Studio Maxx to create the model. The VW is playable online via the Unity web-plugin, or downloadable and run as a stand-alone exe. program (PublicVR, 2008).

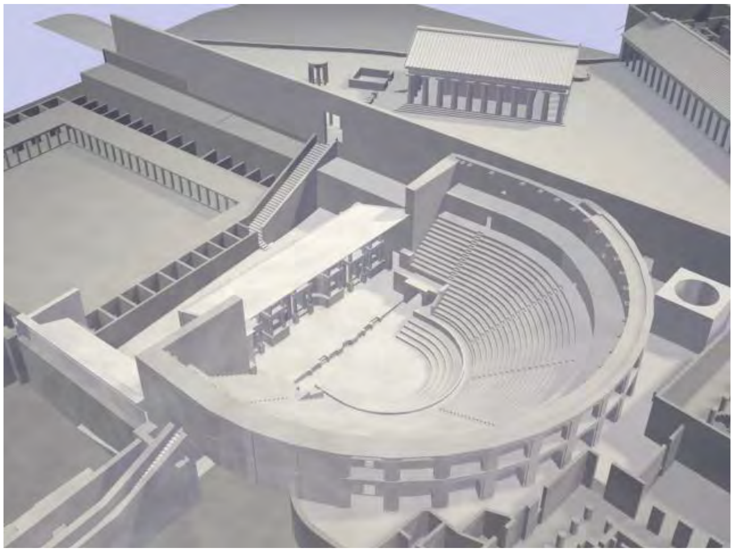


Figure 3.13: Theatre District model in Unity. From (Weis et al., 2010).

3.7.6.3 Useful design and data components

There is no user interactivity available in this VW and all that is offered in navigability through the reconstructed models of the site.

Chapter 4

Identification of input data and design elements in a VW

The focus of this chapter is on the identification of techniques employed in the creation of a VW that enables a user to explore an environment virtually, in an immersive and engaging manner. The ‗useful design and data components‘ identified in the previous chapter from case studies of VH applications are explored here and their implementation in a VW is discussed.

When designing and creating a VW there are two categories of components to consider. The first category consists of the data to be incorporated into the VW and includes 3D models, terrain, characters and photographs. The second category is the design and functionality of the VW and includes the navigation system used to explore the environment, menu buttons, and the layout of the screen. These two categories have thus been labelled as input data, and design elements.

4.1 Input data

To create a VW of a heritage site spatial and contextual data is required to populate the virtual environment. The input data in this study will be limited to spatial data, as educational data, such as historical texts or audio, falls outside of the scope of this research, as described in the limitations section in section 2.9. Spatial data available for heritage sites typically includes the following list of data and will form the basis of the VW. Each element is discussed below.

- Terrain and orthophoto
- 3D textured models of heritage structures
- 360-degree panoramas
- Elevations, sections and plans
- Avatars
- Vegetation
- Virtual objects

4.1.1 Terrain

Terrain is the horizontal and vertical dimensions of the surface of land. These dimensions can be represented by contours, a triangulated irregular network (tin) surface model, or a grid of 3D co-ordinates etc. These representations either form a Digital Terrain Model (DTM) or Digital Surface Model (DSM). A DTM represents the bare ground of the land while a DSM includes vegetation and trees (Bandara, 2011).

Terrain data is essential to a VW as it forms the base of the environment that the virtual visitor will be navigating through. The terrain should be of a high resolution to provide maximum detail and textured with a high resolution aerial or satellite image to achieve a realistic appearing virtual environment.

Terrain data can be generated from a variety of sources which include topographic surveys, aerial and satellite mapping, and lidar. It can also be artificially created if no data is available. Methods of acquiring terrain data often result in the creation of a DSM, where vegetation and buildings are included in the model. Landscapes with rugged mountainous terrain can be difficult to model accurately as steep cliff faces and areas in dark shadow are difficult to record. This problem can be addressed, albeit not always entirely resolved, by choosing appropriate data acquisition methods. Ground-truthing should be carried out on terrain models to correct any inaccuracies. This can be achieved by visual inspection of the site via photographs or on-site inspections. If erroneous terrain data is identified via ground-truthing these areas must be manually modified to preserve the terrain‘s authenticity (Bandara, 2011).

To create a realistic appearance for the terrain model, a high resolution orthographic aerial image should be draped over the terrain. Where aerial or satellite photography is not available to produce an orthophoto some form of artificial texturing must be used to provide texture to the terrain.

Figure 4.1: Aerial view of the reconstructed city centre of Rome showing buildings on top of a terrain model draped with aerial photography. From (Frischerconsulting, 2013).

4.1.2 3D models

A 3D computer model, in the form of geometric primitives or a polygonal mesh, of a heritage structure can be generated from a variety of methods, each with their own strengths and weaknesses. 3D models can be created by virtual reconstruction based on drawings or plans of a structure, or the physically remaining structure can be digitally recorded as a point cloud and subsequently modelled. Where a heritage site structure is in ruins or where only foundations remain buildings can be artificially modelled and virtually reconstructed to resemble their original form as best as can be established from remaining archaeological or historical evidence. Virtual reconstructions often make use of geometric primitives, such as planes and cylinders. This has the advantage of very low polygon counts due to the large, low detail, polygons used in the reconstruction process, which makes viewing and navigating faster and easier. Such models can then be visually improved by draping a sample texture over the surfaces (DeLeon and Berry, 2000). This method is the crudest form of a creating a 3D model of a structure as it lacks the spatial textures of the real world with its uneven walls and irregular shapes. This method typically results in an artificial and unrealistic appearance of the model to the person viewing it. Figure 4.2 shows an example of a reconstructed building with artificial sample texture.

A more authentic and realistic method of creating a 3D model of a structure is to use laser-scanning and/or photogrammetry to record the physical dimensions of the structure by generating point clouds that represent the site. These point clouds are then surface modelled using meshing algorithms and subsequently textured with images taken of the structure. These methods are able to capture detail at high resolution (up to 1mm point interval) and high accuracy (2mm) and offer the most objective and visually appealing solution to digitally preserving heritage sites (Rüther et al., 2011). The inclusion of the resulting 3D models into a VW produces a virtual environment that best represents the real world environment. The drawback of these models is that they are very time and skill intensive to create. For their creation, laser-scanned point clouds must be cleaned of unwanted objects, scans must be registered together, and finally a surface mesh must be generated based on the registered point cloud. The resulting model must then be hole-filled and accurately textured (Held, 2012). The final product has to then undergo intelligent polygon-decimation before being incorporated into the VW as the many polygons making up the detailed meshed model put strain on end-users computer processing power.

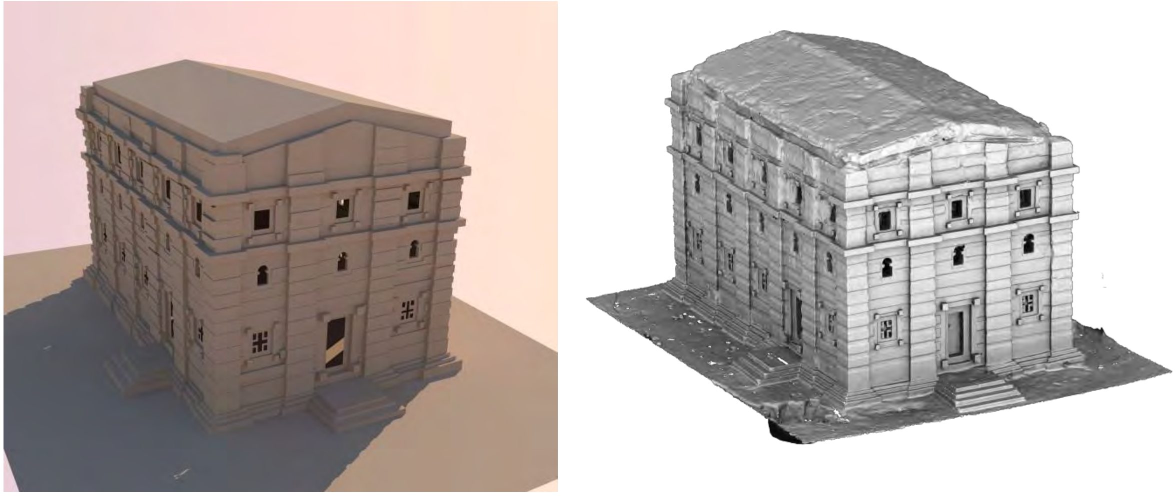


Figure 4.2: Left: Example of a 3D model reconstructed from primitives. Right: A 3D model created from meshing laser-scan point clouds.

For the purpose of a VW the models must be hole-free to avoid the user seeing or falling through the model as this would break the sense of realism in the VW. Where structures have been digitally recorded using laser-scanning or photogrammetry this involves artificially creating data where no data has been captured. Holes exist in the model due to areas of the recorded structure being occluded from the capturing device. For example, in the case of laser scanning, the area directly beneath the scanner will be void of data as the laser beam cannot ‗see‘ this area. Thus the final registered point cloud often has ‗holes‘ where no data has been recorded. Hole-filling algorithms can be used to interpolate data to fill in the holes when the point cloud is meshed to create a model. The models will then be more visually appealing, but less authentic.

4.1.2.1 Texturing

Meshed 3D models are typically shaded with a neutral grey colour. To achieve a realistic visual display the models can be coloured, which is referred to as texturing. Models can have colour directly painted onto their surface, be textured by a repetitive sample texture (as seen in figure 4.3 below) or be textured by images captured of the structure. The first two methods significantly improve the appearance of a standard grey scale coloured 3D model but are generally seen as unsuitable in the holistic, objective documentation of cultural heritage sites (Held, 2012). Texturing is crucial in creating realistic looking 3D models of heritage structures and landscapes. An added advantage of texturing is that a high resolution texture can mask the effects of low resolution (polygon count) models and create models with a realistic appearance.

Figure 4.3: Fallen stele of Axum, Ethiopia. Left: Real colour. Right: Sample colour.

The third listed method, the ―draping” of images over the model surface, requires that photographic imagery of the structure is captured. This imagery is then applied to the model by using either vertex colouring, or texture mapping (Ranzuglia et al., 2012)). Both methods are discussed in detail below. To acquire colour information of a structure many laser-scanning instruments are equipped with on-board cameras and capture photographic imagery as a separate step to laser scanning. The advantage of this method is that all surfaces captured via laser scanning will have colour information and that all images are automatically aligned to the scanned scene. However, the disadvantage is that the imagery is captured at the time of the laser scan which can introduce differences in the appearance of the scene depending on the suns position at the time of day. Long scanning sessions which can take the course of a whole day will thus produce imagery that is widely varied in lighting condition as different areas of the structure being scanned are lit up differently throughout the day. Using hand-held cameras to capture imagery at a similar time of day is thus considered the most desirable approach.

The captured imagery must then be projected onto the meshed 3D model. This requires the alignment of the photograph to the 3D model and is achieved using basic photogrammetric principles where the internal and exterior orientation parameters are known or determined. This alignment can be achieved in a number of ways, either by aligning individual images to the model or aligning images to each other and then aligning this set of images to the model. Both methods require the use of common points on the model and the image (Held, 2012).

Once the images are aligned there are two methods, as mentioned above, of creating the texture for 3D models; vertex colouring and texture mapping. Vertex colouring assigns a colour value to each vertex of a triangle of a meshed surface model, and interpolates the colour for the surface between each vertex. This method provides a simple way to store colour in a 3D model, but the detail of the texturing is dependent on the resolution detail of the underlying mesh.

Texture mapping on the other hand can provide a very high resolution texture to a low resolution model, by linking an area on a texture atlas image to each triangle on the model. This method is known as parameterisation, whereby 2D image coordinates, also known as UV co-ordinates, are assigned to each vertex of a triangle. Parameterisation is optimised by having as many triangles lying next to each other on the texture atlas image as possible to allow for manual editing of the atlas in photo manipulation software. In this way dark areas of the texture atlas, and by extension the textured model, can then be brightened. A high resolution texture atlas can be linked to a low resolution model and provide high quality textures and a good visual appearance to an otherwise visually unappealing model.

Figure 4.4: Parameterisation of a hand Axe found in Cape Town, South Africa. Left: 2D texture atlas image. Right: Textured 3D model using the texture atlas.

4.1.3 Elevations, sections and plans

Elevations, sections and plans are 2D projections of a 3D model displayed on an image. See figure 4.5 for examples of each. Elevations are orthogonal, vertical views of a 3D model shown from the outside of the model. Sections are similar views, but the model is cut vertically to show the inside of the model. Plans are orthogonal horizontal cuts through the 3D model to show ground, roof or intermediate levels. Elevations, sections and plans include scale and orientation on their layout and can be used for acquiring measurements of the structure. To create objective representations the 3D model should be metrically accurate, have high resolution detail, and not be hole-filled by artificial interpolation (Zamani Project, 2015). Elevations, sections and plans allow a user to gain a perspective of the 3D model of the structure that would not be possible in real-life situations. Added to this is the ability to measure and view elements of the structure in relation to each other, making it possible to extract metric information about the documented structure.

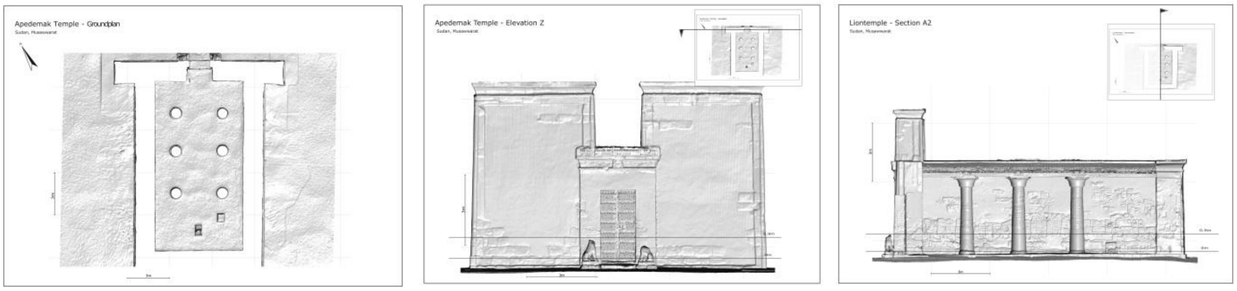


Figure 4.5: Example of plans (left), elevation (middle) and sections (right) of the Apedemak temple, Musawwarat es-Sufra, Sudan.

4.1.4 Panoramas

360-degree full-dome panoramas are images that, when viewed in appropriate software, enable a user to see a full spherical field of view surrounding the panoramas position. They create the impression of standing on a location with a view available in all directions. There are different methods of capturing panoramic images including using a wide angle lens, whereby the full-dome view is captured in sequence and the individual images are stitched together, or by using a camera setup with a curved mirror attached to the front of the camera that reflects a 360-degree field of view into the camera lens.

Panoramas and panorama tours, as described in section 3.2, are an effective method of immersing a user into a heritage site scene. The user can pan and zoom in the panorama and experience a very realistic feeling of being in a specific spot. Panoramas can be integrated into VWs allowing a user to jump to chosen panorama positions by clicking panorama positions on a map or in the virtual environment scene (Osman, 2009).

Figure 4.6: Field of view of a panorama. Example from inside the church of Bet Medhane Alem, Lalibela, Ethiopia.

4.1.5 Avatars

Avatars in the form of computer generated animated characters augment the physical environment by introducing dynamic historical figures to the user's experience of the site. These characters would be dressed in historical garb and include historical and religious figures, servants, kings, herdsmen etc. Additionally the avatars can be equipped with weak artificial intelligence (AI) to create a bot that can act as tour guides that lead the user through the site and offer the user audio narration of educational data about the site.

Figure 4.7: Avatar from Vari House. From (Jacobson and Sanders, 2013)

4.1.6 Flora

Artificial vegetation can be added to a virtual scene to create more lifelike environments that provide the user with a sense of the climate and the environmental conditions of the site. Dry desert-like VWs would have little vegetation with palm trees and small shrubs being the exception. More leafy vegetation can be added to scenes in tropical climates. Flora can also be equipped with scripts that create the impression of the vegetation moving in a breeze.

Figure 4.8: Flora in a Unity virtual environment.

4.1.7 Virtual objects

Virtual objects can be placed in the scene as dynamic interest points. These objects would be historical artefacts such as vases, swords, cooking equipment etc. The objects can be made interactive by making them clickable, collectable and/or movable. They offer an insight into the culture of the people who inhabited the site.

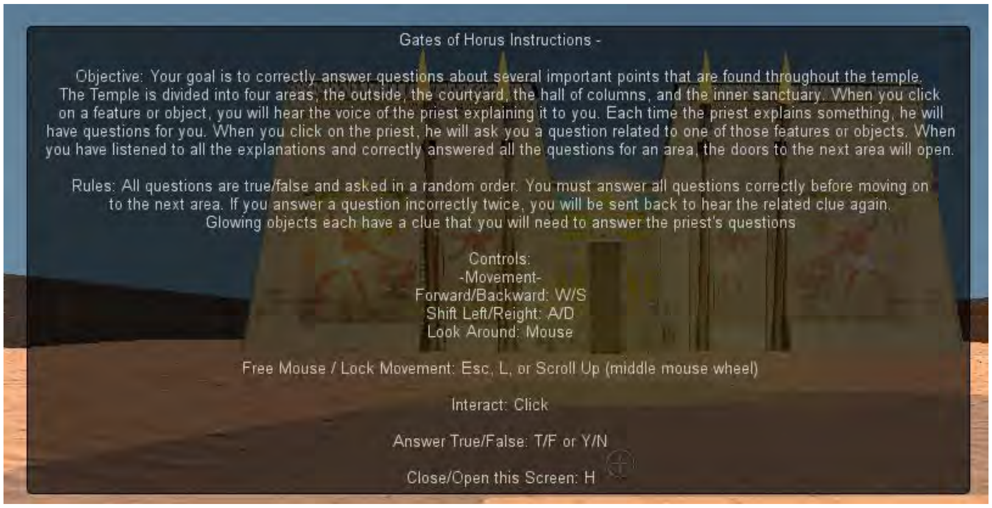
Figure 4.9: A virtual object. From (Jacobson et al., 2009).

4.2 Design elements

From case studies of the virtual heritage applications available today the following design elements have been identified and should be implemented to make a VW immersive and engaging for a user.

4.2.1 Start screen

An introductory start screen should be available for the user to access some basic information about the VW including instructions on how to navigate, jump to new locations, access any virtual tour guides or play educational games. The start screen should explain to the user the purpose of the VW as well as a background history of the site and copyright ownership of the data.



Gates of Horus Instructions -

Objective: Your goal is to correctly answer questions about several important points that are found throughout the temple. The Temple is divided into four areas, the outside, the courtyard, the hall of columns, and the inner sanctuary. When you click on a feature or object, you will hear the voice of the priest explaining it to you. Each time the priest explains something, he will have questions for you. When you click on the priest, he will ask you a question related to one of those features or objects. When you have listened to all the explanations and correctly answered all the questions for an area, the doors to the next area will open.

Rules: All questions are true/false and asked in a random order. You must answer all questions correctly before moving on to the next area. If you answer a question incorrectly twice, you will be sent back to hear the related clue again. Glowing objects each have a clue that you will need to answer the priest's questions

Controls:
-Movement-
Forward/Backward: W/S
Shift Left/Reight: A/D
Look Around: Mouse

Free Mouse / Lock Movement: Esc, L, or Scroll Up (middle mouse wheel)

Interact: Click

Answer True/False: T/F or Y/N

Close/Open this Screen: H

Figure 4.10: Start screen from the Gates of Horus educational game.

4.2.2 Menu options

Menu options can give a user a choice of what is displayed on screen etc. There are no specific menu options that should be included in each VW. The menu should be easy to use so that users can intuitively explore the options themselves.

4.2.3 First person/third person view of world

There are two standard options of viewing and navigating through a virtual environment scene in 3D: first and third person perspective. A first person perspective enables users to experience the VW as if they were physically walking in the scene and thus the environment is viewed as if through their own eyes. A third person perspective is achieved by enabling the user to control an avatar that moves through the scene while viewing the avatar walking in the environment. Third person perspective viewing thus has the additional requirement of creating a lifelike avatar which induces users to feel they are exploring the site as if they are the avatar they control (Svånå, 2010).

4.2.4 Navigation

Users should have access to the same navigation abilities in the VW as they would in real life. Thus the user controls must enable the user to move forward, turn and view in a full 360-degree motion. Additional navigability includes jumping and crouching. The controls used to control the movement of the visitor through the scene are typically the mouse and keyboard, though other methods, such as joysticks, can also be used.

4.2.5 Mini-map

A mini-map is a top-view orthographic rendering of the VW scene that shows the layout of the scene and the location of the user within it. Typically the mini-map is displayed in a corner of the screen, is centred on the user and shows the users current view direction. The mini-map usually displays a small portion of the scene around the user and can be zoomed in or out to gain a perspective of the entire scene. The mini-map often displays icons indicating points of interest or other functionality that exists at specific locations. The icons can be clicked to activate their functionality and often have tooltips associated with them to indicate what their function is.

Figure 4.11: Mini-map from World of Warcraft by Blizzard Entertainment. Player's current location and orientation can be seen with nearby points of interest. From (Svånå, 2010).

4.2.6 Additional pop-up information

To provide the user with additional information about the virtual heritage site there should be access to information panels, audio narration, short videos, and photographic views. An example of pop-up data would be a virtual display panel that a virtual visitor could walk up to and then click on to view in full-screen. Information to be displayed on these panels could include elevations and plans of the structure. Other functionality could enable a user to jump into a 360-degree real world panoramic view of the user's current location. To draw attention to the pop-up data a floating icon should hover over the display with a tooltip describing its function. A glowing hue could also surround the feature.

Figure 4.12: Example of pop-up data from Gates of Horus. A yellow glow surrounds the statue on the left indicating the statue can be clicked on to display more information or functionality.

4.2.7 Audio

Audio plays an important role in setting the mood of a scene. Audio can capture the ambiance of a place and creates a feeling of presence. The unique sounds that are inherent to a heritage site offer an alternative sensory input into the VW other than sight. Audio can also include narration by avatars to give instructions to the user if the VW includes an educational game, or to communicate information about the site.

Chapter 5

Implementation

This chapter discusses the design and creation of the VW of Petra. As described in chapter 4 the development of the VW has been split into two implementation groups: Creating and importing the spatial data into the VW - shortened to ‗Input data‘, and ‗Design elements‘. This section introduces the software toolkit that was chosen for the development of the Petra VW before discussing the design and creation of the VW. Refer to Appendix A for instructions on how to access and run the VW of Petra.

5.1 Choice of software toolkit

To fulfill the objectives of this research a software toolkit has to be chosen that can integrate all the input data and design elements as detailed in chapter 4. To build such a toolkit from the ground up would require an extensive investment in time and money and would be the equivalent of ―re-inventing the wheel”. As described in Chapter 3, game development toolkits offer the functionality of creating virtual environments and supply all the tools needed to realise the goal of an interactive VW with real-time, high quality graphics rendering capabilities. It was thus an obvious decision to adopt a game development toolkit for the development of the Petra VW.

A software package had to be chosen that is available at no cost, requires only a beginner level of programming skills and makes it possible to incorporate all the identified VW components. Important criteria in the choice of the software were the ability to create an interactive first person perspective view of the virtual environment with user control and navigation capabilities. Also the VW had to run on standard computers and be available on a web-browser.

The software chosen to create the VW of Petra was Unity. The comparison Table 1. shows that Unity offers all the requirements as discussed above. The virtual heritage applications discussed in chapter 3 which use Unity in their development showed very good results which proves Unity is a very capable software platform. Unity has been developed to be as easy to use as possible to enable beginner developers with minimal coding experience to create realistic virtual environments. Also Unity uses Java as one of its scripting language which the author is familiar with and since there will be extensive programming required to create the VW it is best for the author to program in a language that he is familiar with. Refer to Appendix B.1 on how Unity is used to create a VW.

5.2 Data acquisition and processing

The spatial data of Petra was captured during five field campaigns to Petra between April 2011 and June 2014. Each field campaign comprised of roughly 5 – 8 days. Approximately 2200 laser-scans, 450 panoramas, one hundred and fifty RTK GPS measurements, and fifteen thousand images for texturing were captured during these campaigns. Extensive GIS datasets, available from previous research and documentation projects, were imported from multiple sources and merged with newly created data into a single GIS based on a common reference system, UTM 36N on WGS 84. There were also 15 aerial images of Petra which were used to create a DTM and orthophoto of the site. All newly captured data was processed by the Zamani project team, which is described in detail by Rüther et al. (2014), at the University of Cape Town. Most important for the VW is a point cloud of 3 km length which was created by registering the 2200 acquired laser-scan data together, thus creating a single point cloud which includes all major heritage structures as well as the surrounding terrain. This point cloud was geo-referenced to the GPS points via points which could be identified in the point cloud and surveyed by GPS measurement. On screen digitising on an orthogonal projection of the point cloud served to contribute topographic data and other features to the GIS.

The output of the processed data of Petra can be seen in figure 5.1 below and includes:

- 3D models of 30 major structures, most of these textured

- 3D model of the 1.2 km long Siq (the access canyon to Petra)
- Plans, sections and elevations of each 3D model
- 450 panoramas in panorama tours
- DTM and orthophoto of Petra and surrounds
- Comprehensive GIS with historical and current data

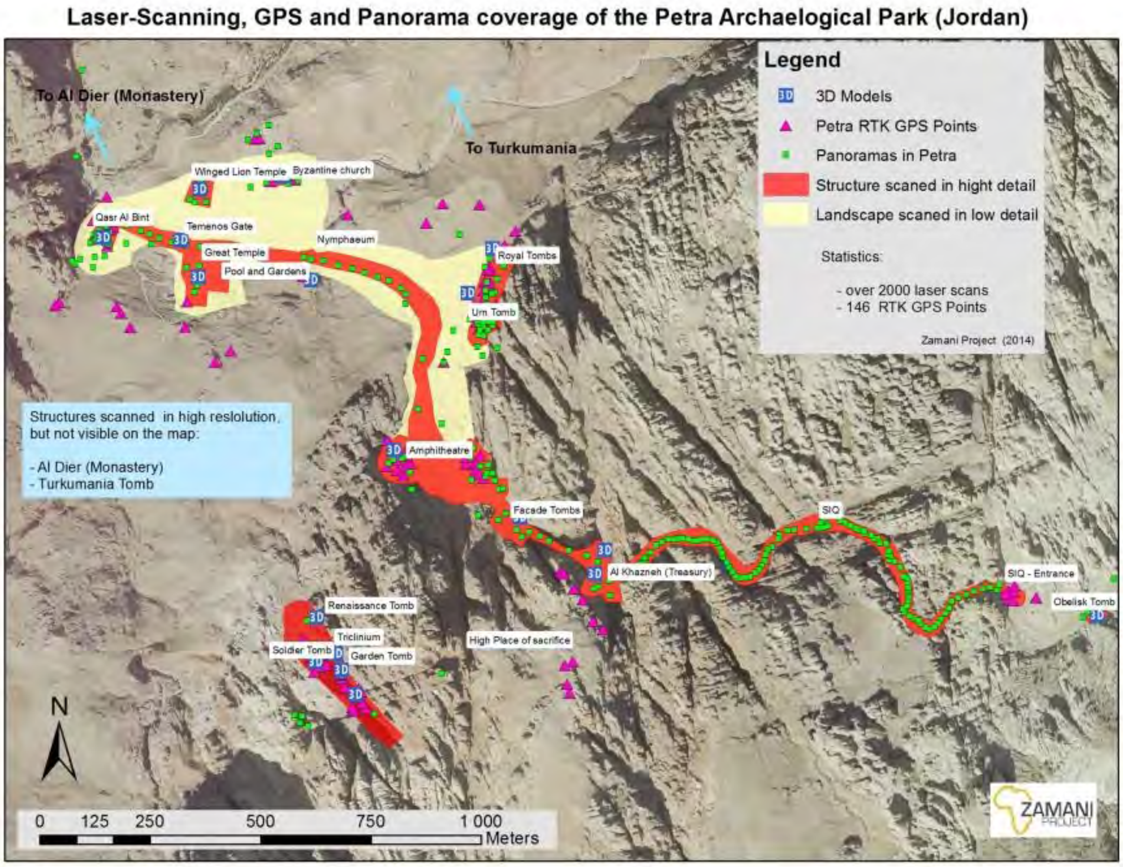


Figure 5.1: GIS layout of spatial data captured in Petra by the Zamani team.

5.3 Creation of spatial data for the Petra VW

In order for all spatial data generated of Petra to fit into the VW in a seamless manner all the data was geo-referenced to the co-ordinate system UTM 36N on WGS 84. By using this approach the terrain and 3D models align exactly to each other inside Unity and no manual placement of virtual objects in the virtual environment scene is necessary when importing this spatial data.

The following spatial data was added to the VW of Petra.

5.3.1 Terrain

A 1m grid of DTM points, covering an area of 20 square kilometers, was created from 15 aerial images and 50 RTK GPS points and can be seen in figure 5.2 below. Approximately 0.35 square kilometers of terrain was captured by terrestrial laser-scans captured on site. These scans add detail to the terrain created by the aerial DTM (this is discussed in detail futher in this study). An orthophoto with a 0.2m pixel resolution was also created from the aerial images.

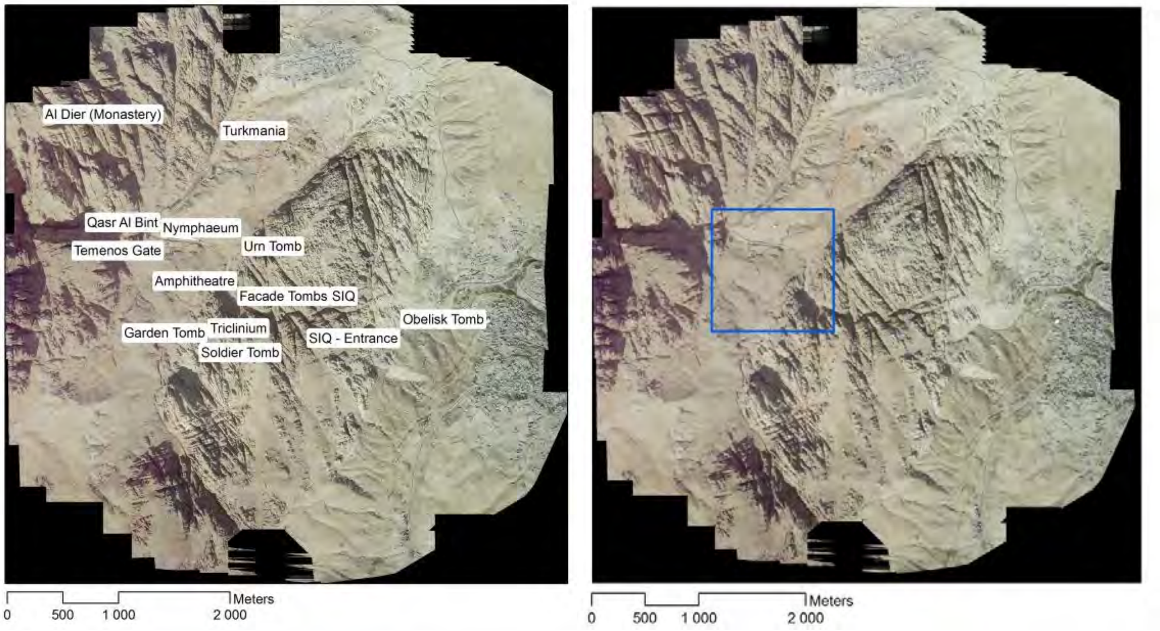


Figure 5.2: Left: Image of the entire DTM and orthophoto of Petra. Right: The terrain area chosen for the VW is highlighted by a blue square.

The extent of the terrain to be included in the VW had to be chosen to include as many of the important monuments of Petra as possible. However, it was not possible to choose the entire terrain of Petra for inclusion since there is a limit, set by Unity, of how much detail can be displayed of the terrain. This limit determines that the amount of detail displayable of the terrain is inversely proportional to the size of the terrain. This is discussed in detail below. Therefore, the smallest area of terrain that includes the most monuments was chosen and can be seen in figure 5.2 marked by a blue square.

he important monuments of Petra, namely the Treasury, Amphitheatre, Royal Tombs, Great Temple and Qasr al Bint can be visited following a linear path (see red line in figure 5.3 below). However, the terrain for this area of Petra had to cover a rectangle (as Unity requires rectangular areas of terrain) incorporating these monuments and so a 1.1km x 1.1km area of terrain was chosen for this function.

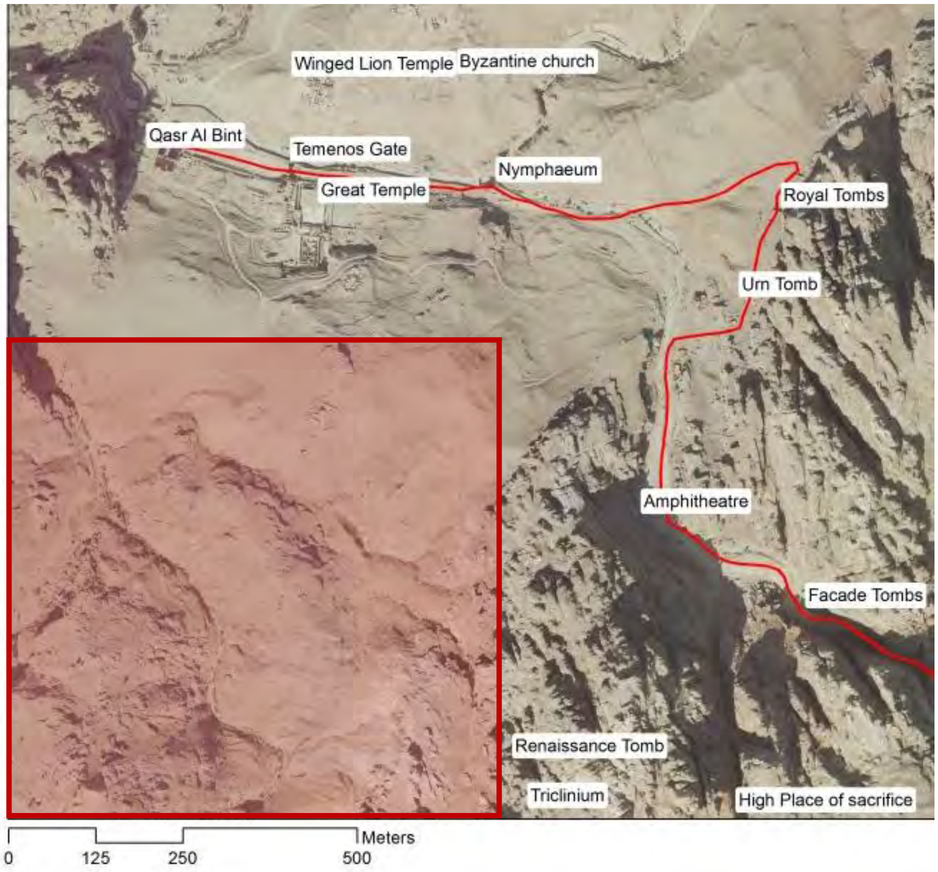


Figure 5.3: Red line shows linear path of the important monuments of Petra. Red square highlights unimportant terrain void of monuments.

This enforced choice of terrain coverage leads to large portions of terrain that the user is unlikely to explore as indicated in the red area in figure 5.3. The level of detail of the terrain has to remain consistent (as required by Unity) and so a large area of unused detailed terrain will take up valuable computer processing resources. This is unfortunately unavoidable.

Besides the chosen terrain areas described above, there are two other important areas of Petra that should be included in the VW, and a decision had to be made regarding the terrain used for these areas. The first of these areas is the Monastery, a carved Nabataean monument similar to the Treasury, which is situated 1.1km North West of Qasr al Bint. The other area, Wadi Farasa with four important Nabataean monuments, besides numerous other tombs and structures, lies 500m south of the Amphitheatre.

As the Monastery is geographically isolated, with no other important monuments nearby, terrain data is not necessary for this area. The user is not expected to explore beyond the extent of the Monastery, for which a laser-scan 3D model exists. An artificial, flat terrain was therefore created in Unity to surround the 3D model.

Wadi Farasa consists of a number of monuments that are carved into the side of a valley (Wadi) and is approximately 500m long and 50-100m high. The landscape and monuments of this area were extensively recorded with over 200 terrestrial laser-scans. A landscape 3D model which included the monuments could be created from this laser-scan dataset to represent large sections of Wadi Farasa and thus a terrain dataset generated from the aerial imagery was not required for this area.

5.3.1.1 Merging the aerial DTM and terrestrial laser-scans to create the terrain

The 1m grid points generated from the aerial DTM representing the entire terrain of Petra were meshed (a meshed surface represented by triangles (Rüther et al., 2012)) to produce a DTM of 1m resolution. It was found that the surface model was inaccurate in areas of the terrain where there are steep rock faces, and completely unreliable where there are dark shadow areas in deep ravines in the aerial images used to create the DTM. Since the landscape of Petra consists of many steep rock faces and large areas in shadow as a result of these steep faces much of the aerial DTM data is unreliable.

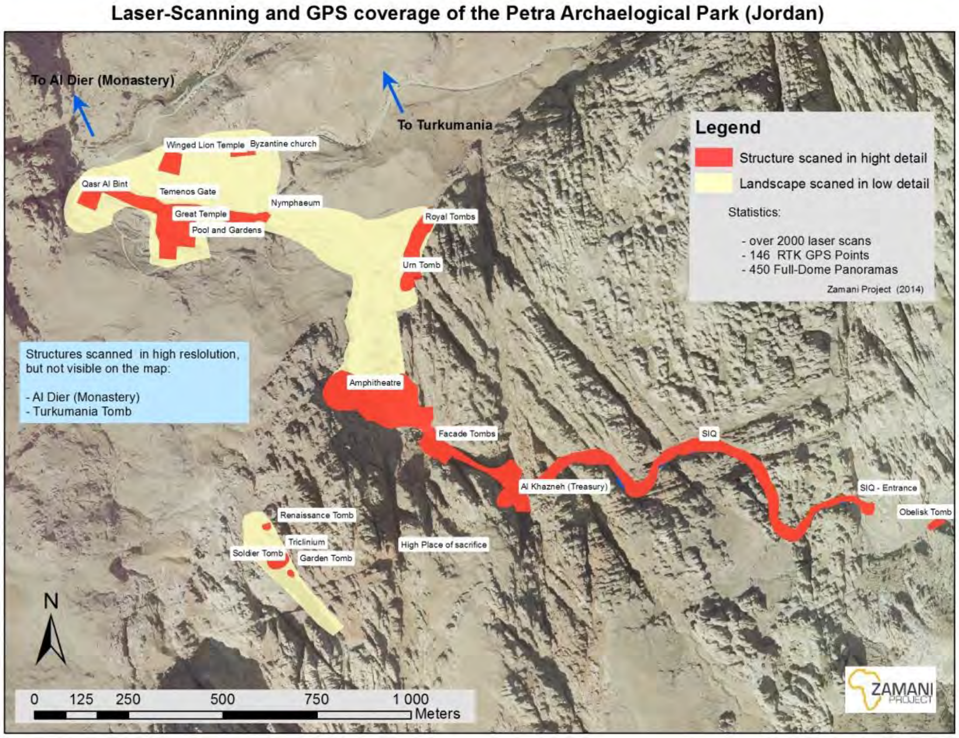


Figure 5.4: Laser-scan coverage of Petra.

As indicated in figure 5.4 above the red and yellow areas show the coverage of the 0.35 square kilometer area of terrestrial laser-scan data of Petra. This laser-scan point cloud was surface modeled and decimated to an average of 30cm resolution to create a laser-scan landscape 3D model. There are no doubts about the accuracy of this model, however, small areas within the laser-scan landscape model were occluded due to limited line of sight from laser-scanner positions to rock faces and these resulted in holes in this model as can be seen in figure 5.5.

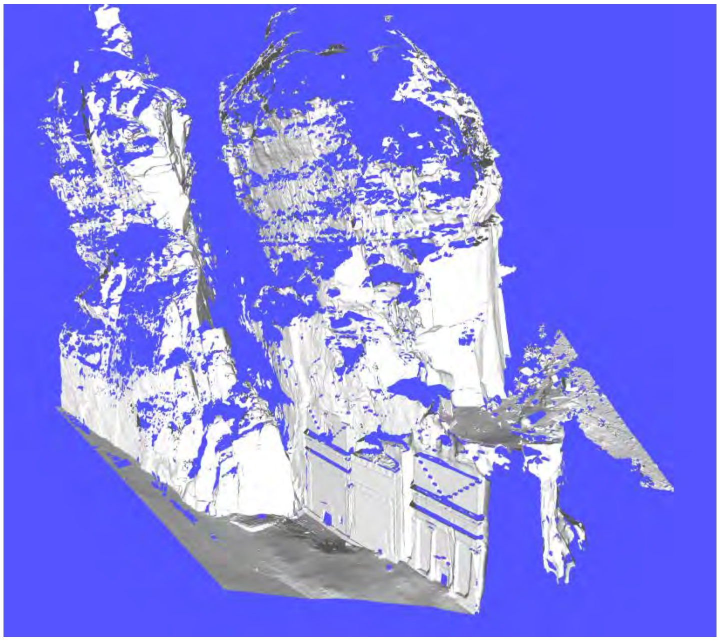


Figure 5.5: Example of occlusion holes in landscape model.

To create the most accurate representation of the terrain possible it was decided to merge the two sources of terrain data together, giving preference to the laser-scan landscape model where it is available (see figure 5.6). The area covered by the laser-scan landscape model was then cut out of the larger area covered by the 1m-resolution aerial DTM and replaced by the laser-scan landscape model. The two DTMs were joined together using a hole-filling meshing algorithm to achieve a seamless merge between the different resolutions This resulted in a merged terrain which combines the high level of detail of the laser-scan data with the larger, but lower level of detail, coverage provided by the aerial DTM.

Refer to Appendix B.2.1 on how terrain data was imported into Unity.

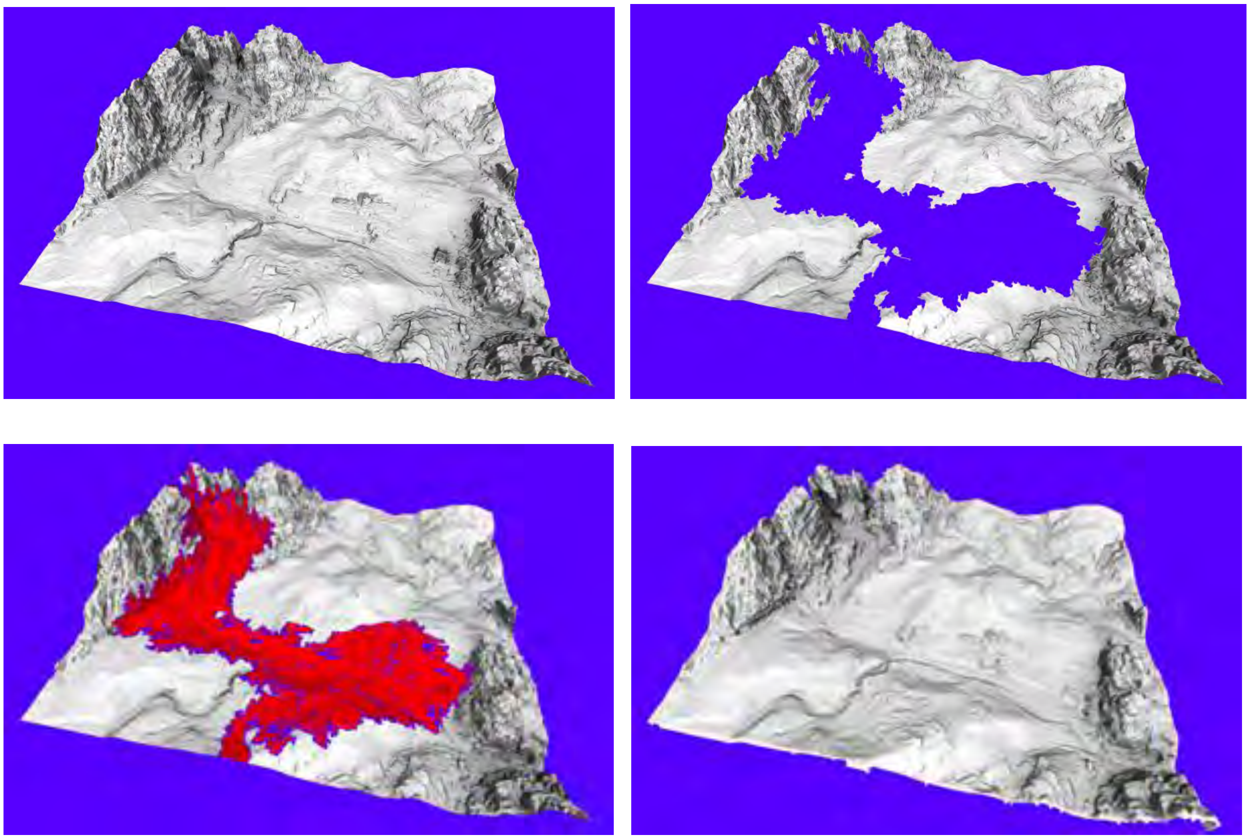


Figure 5.6: Top left: Aerial DTM. Top right: Aerial DTM with laser-scan landscape area removed. Bottom left: Laser-scan DTM, shown in red, added to the aerial DTM. Bottom right: Combined laser-scan and Aerial DTM.

5.3.2 Orthophoto

The 20cm orthophoto created from the aerial images was cut to the same size terrain chosen for inclusion in the VW and imported into Unity to be used as a texture to colour the terrain as can be seen in figure 5.7 .Unity allows images of a maximum size of 4096 x 4096 pixels to be imported. Since the terrain has a length of 1100m this calculates to a pixel size of 0.3m (1200m/4096pixels).

The orthophoto was also used as a back-drop for the mini-map, described below, which functions as a small scale GIS of the site. Refer to Appendix B.2.2 on how the orthophoto was imported into Unity.

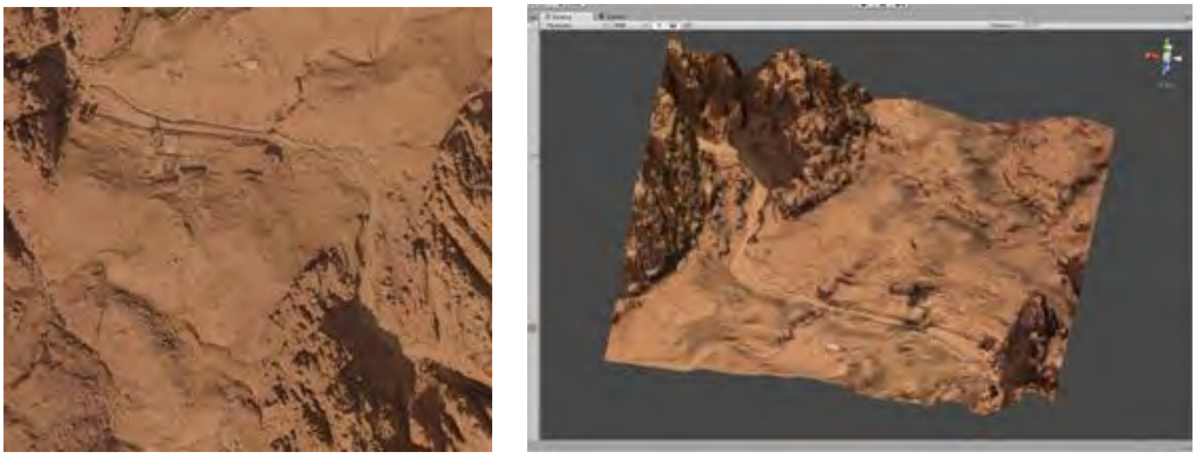


Figure 5.7: Left: Orthophoto of Petra. Right: Orthophoto draped over Petra terrain in Unity.

5.3.3 3D models and texturing

5.3.3.1 Laser-scan data capture

As a partner in the Siq Stability project, described in section 2.1.2, the Zamani team was tasked with the digital documentation of the important monuments of Petra. The exact number of monuments to be documented was not specified at the beginning of the project as there are over 1500 tombs situated throughout the Petra Archeological Park. The Zamani team aimed to record as many monuments and as much landscape as was possible in the five field campaigns that took place.

The method of documentation chosen to record the monuments was primarily laser-scanning, as opposed to photogrammetry. Laser-scanning offers the ability to record physical structures with very high accuracy and is not subject to light conditions and target placements (geometric surface scan registration was used) that can hinder photogrammetry. The team used a variety of laser-scanners in Petra which included: Z+F 5010, Z+F 5010c, Z+F 5006, Leica HDS 3000, Leica HDS 6100, Leica C10, and Trimble FX. These scanners offer differing ranges, from 40m to 150m, different accuracies, scan times, and colour capture capabilities.

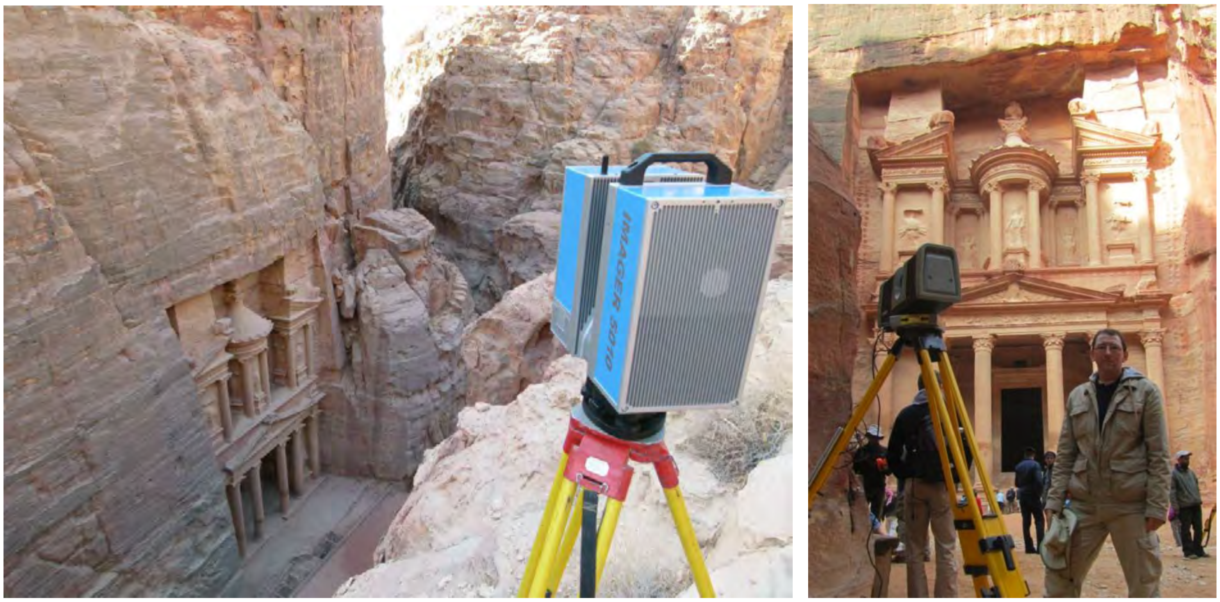


Figure 5.8: Images of Zamani team laser-scanning in Petra.

These individual laser-scans were then processed into individual 3D surface models of monuments and landscape using a pipeline developed by the Zamani project (Rüther et al., 2012). The process involves scan cleaning, registration, modeling, hole-filling, texturing and finally geo-referencing the models to a chosen co-ordinate system.

5.3.3.2 Cleaning of Petra laser-scans

All laser-scan point clouds captured in Petra had to be cleaned of unwanted objects such as people, vegetation, furniture, cars and donkeys. Scan cleaning is a time intensive, manual process, but if not carried out results in erroneous polygons in the final 3D model which detracts from the appearance and authenticity of the model. The Zamani policy on cleaning scans is to remove any objects that are not historically relevant or not natural to the site.

The cleaning task was made difficult by the large number of tourists, which can reach up to 2500 a day, that visit Petra. These tourists often walked through the line of site of the laser-scanner to the targeted rock surface area being scanned thus introducing erroneous points into the scan point clouds as can be seen in figure 5.9 below. Additionally, they occlude object areas which results in holes in the point cloud where objects should have been captured. The obvious solution to this problem would be to restrict tourists from entering an area that is being laser-scanned. Unfortunately it was not possible to do this in most cases as the tourists walk a predefined path through Petra and halting them in the path is not practically possible. However, where possible, measures were taken to keep people away from the scanner while scanning.

Figure 5.9: Images of people walking in front of the laser-scanner.

5.3.3.3 Registration of laser-scans

As mentioned above, the individual laser-scans were registered together using geometric surface feature matching as well as ICP algorithms as opposed to target based registration. The reason that this approach was chosen was because placing targets throughout the site would have been extensively time-consuming, and impractical. As each scan needs to capture a minimum of three targets it would have been logistically very difficult to place enough targets for the 2200 laser-scans captured. Additionally, the protocol of a World heritage Site protects against disturbing the heritage monuments in any form, which would have been unavoidable by placing targets. It is policy in the Zamani project pipeline to register scans based on geometric surface matching (Rüther et al., 2012).

All scans were subsequently combined into a single, registered point cloud. This point cloud was then geo-referenced by a best fit transformation without scaling, to high accuracy (+-2cm) GPS points distributed throughout the site. GPS points were captured on features, such as corners of structures, which could be subsequently identified in the combined point cloud. The accuracy of the best fit of the point cloud to the GPS points was approximately 0.3m over 3km, which was deemed highly satisfactory.

5.3.3.4 3D surface Modeling

The individual heritage monuments of Petra were surface modeled using meshing algorithms. The resolution of the final 3D models was typically chosen to be 1-2cm. As a result of this high resolution these 3D models can run into very high polygons counts, depending on the size of the monument, and can consist of tens of millions of polygons per structure. These models were subsequently decimated to be displayable on a standard computer as can be seen in figure 5.10.

For the use of these 3D models in the VW, the models have to be completely hole-filled as any holes in the model would enable the virtual visitor to see through these openings and thus through buildings and walls, which would be unrealistic, and if there are holes at ground level, the visitor could ‗fall‘ out of the model.

A limitation of the Unity software which emerged during the creation of the VW is that individual structures should not have more than 500 000 polygons to guarantee quick loading and smooth running speeds. This makes it necessary to decimate the polygon counts of models significantly from the original millions of polygons to not more than half a million, which unfortunately reduces the displayed detail and causes the models to be visually unappealing (see figure 5.10). To counter this visual effect the models were textured with high resolution texture maps, which add a great amount of visual detail to low resolution models. This is discussed in detail further in the text.

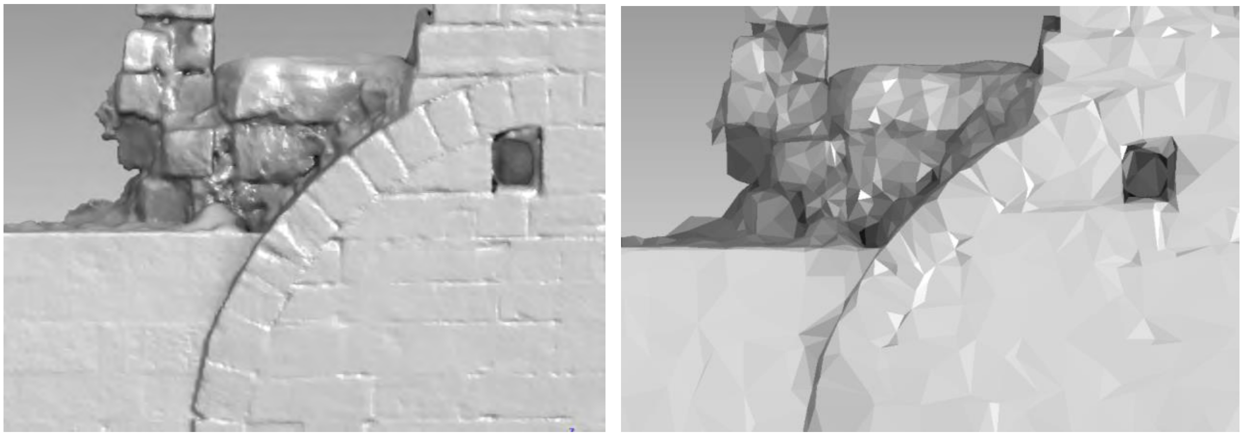


Figure 5.10: Left: High resolution 3D model of Qasr al Bint, Petra. Right: Low resolution model of the same area.

Besides 3D models of individual monuments, landscape models were created for specific areas of Petra. These were meshed with a hole-filling algorithm and included the last 100m of the Siq at the entrance to the Treasury, the area around the Treasury and the connected passage leading to the facades tombs and tomb 835, and the landscape of Wadi Farasa. Significantly lower resolution models were created, in the order of 0.05m – 0.30m resolution, for these landscapes as these models do not need to show great detail. Additionally, modeling of the ground and inside spaces of structures was done with very course detail, as high detail is not needed. Landscape sections that are high above the eye-level of the virtual visitor in the VW and which will never be seen at close range were decimated to its lowest level possible while still retaining the geometry of the landscape. There is thus a varied range of resolutions for the landscape models.

The following is a list of 3D models included in the VW.

	Main Petra Area	**Textured**
1	Exit of Siq	Yes
2	Area surrounding Treasury	Yes
3	The Treasury	Yes
4	Passage from Treasury To Façade Tomb	Yes
5	Tomb opposite the Treasury	Yes
6	Façade Tombs	Yes
7	Tombs Opposite Façade Tombs	Yes
8	Tomb 825	Yes
9	Tomb above 825	Yes
10	Amphitheatre	Yes
11	Urn Tomb	No
12	Corinthian Tomb	Yes
13	Silk Tomb	Yes
14	Palace Tomb	Yes
15	Qasr el Bint	Yes
16	Great Temple	No
17	Temanos Gates	Yes
18	Winged Lion Temple	No
	Wadi Farasa	
19	Landscape of Wadi Farasa	Yes
20	Garden Tomb	Yes
21	Soldiers Tomb	Yes
22	Triclinium	Yes
23	Renaissaince Tomb	Yes
	Monastery area	
24	Monastery	Yes

Table 2: List of 3D models of monuments included in the VW of Petra

5.3.3.5 Texturing of 3D models

As discussed above it is important to attach high detail textures to 3D models to maintain a high visual detail of rock surfaces while keeping polygon counts low. Of the two texturing methods discussed in section 4.1.2.1 UV texture mapping was chosen as the method to use in the case of the VW. Texture mapping enables high quality textures to be draped over low polygon models as can be seen in figure 5.11 below (Held, 2012). Vertex based texturing requires very detailed 3D models with high polygon counts, which is not ideal for this case, and Unity currently only supports texture mapping.

Figure 5.11: Left: Low resolution untextured model of Qasr al Bint. Right: Low resolution textured model.

Texturing the 3D models posed significant challenges. Initially panoramas, as opposed to standard images (images captured with approximately 18mm – 50mm camera lenses), were prioritised for texturing the models as they covered a wider field of view. The panoramas were cubed (divided into 6 parts of a cube) and aligned to the model along with standard images. After inspecting the textured model it was found that areas where two or more panorama cubes or standard images overlapped resulted in a smeared appearance despite the use of filters which allocate weights to images as they are used for texturing. This can be seen in figure 5.12. Weights are based on parameters such as distance from model, angle of incidence and distance to image border (Held, 2012). To reduce the smearing effect, images were masked, thus reducing the overlap to small areas at mask-edges. This resulted in a significant improvement and an approach was adopted in which models are textured with a minimum number of images which focused on specific areas of the model thus avoiding or minimising overlaps. Panorama cubes were found to cover too wide an area resulting in lots of overlap and the original policy to employ panoramas for texturing was therefore abandoned and only individual images taken orthogonal to the surface to be textured preferred. All images used for texturing were manipulated to remove shadows and have similar base colours so that the texturing appeared seamless.

Figure 5.12: Examples of image smearing inside Qasr al Bint.

Besides masking out areas where overlap occurs images had to be masked to remove unwanted artefacts which were captured in the images. This is a similar process to scan cleaning as people, vegetation, furniture and donkeys were all captured in the images. Failure to remove these artefacts results in the artefacts being projected onto the surface of 3D models as can be seen in figure 5.13 below.

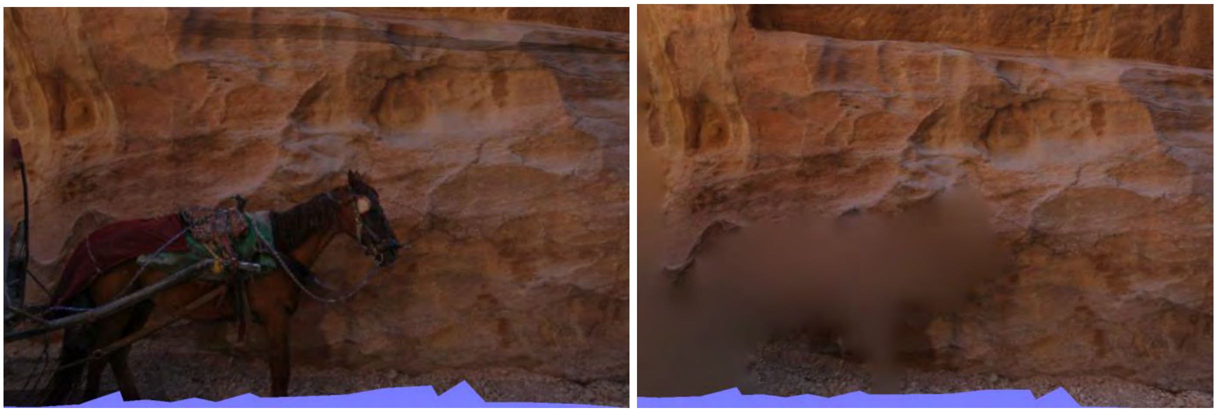


Figure 5.13: Unmasked image projection (left) compared to masked image projection (right).

A complication arose when attempting to cover the large surface/wall areas of the various 3D models with a single UV texture map. UV maps in Unity can comprise of a maximum of 4096 pixels which resulted in an unappealing, low resolution appearance for most and especially larger models since each pixels had to cover too large an area. Structures were thus generally textured using two or more UV maps which involved splitting up the models into smaller parts.

Landscape models were split up into smaller chunks based on what the virtual visitor could see at close range so that higher resolution textures could be achieved for areas at ground level. Areas that are not visible from close range were textured with lower resolution. This can be seen in figure 5.14 below.

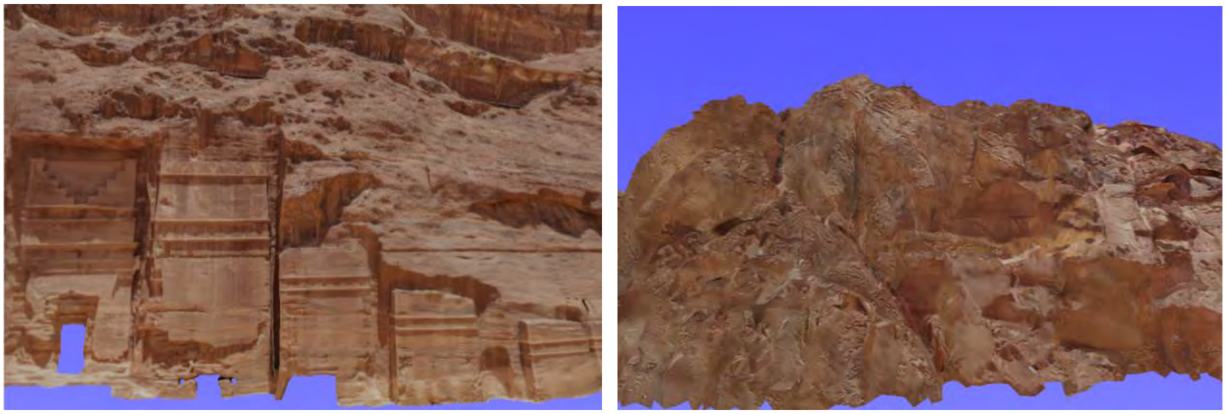


Figure 5.14: Left: High resolution textured model of tombs at eye level. Right: low resolution textured landscape model above eye level.

For texturing of ground and inside of unimportant internal areas a different method of applying texture was chosen. It was not possible to acquire good quality images of the ground since a form of aerial photography would have been necessary. The Zamani team was unable to obtain permission to fly a camera enabled UAV, balloon or kite in Petra. It was also difficult to capture the ground, walls and roofs of internal caverns due to the light conditions inside these structures. Photorealistic details of these surfaces was anyway deemed unnecessary as they add little value to the overall VW.

It was then decided to use a sample texture to texture these areas. Initially a single base colour was applied to these models, but this resulted in an unrealistic appearance since in reality there is never a consistent single colour of a rock feature. To overcome this problem a sample texture was created consisting of a grainy pattern to represent gravel and rock. A generic UV map was then generated for each of these models and the sample texture was applied to the UV map. This resulted in a far superior appearance and can be seen in figure 5.15 below.

Refer to Appendix B.2.3 on how 3D models were imported into Unity.

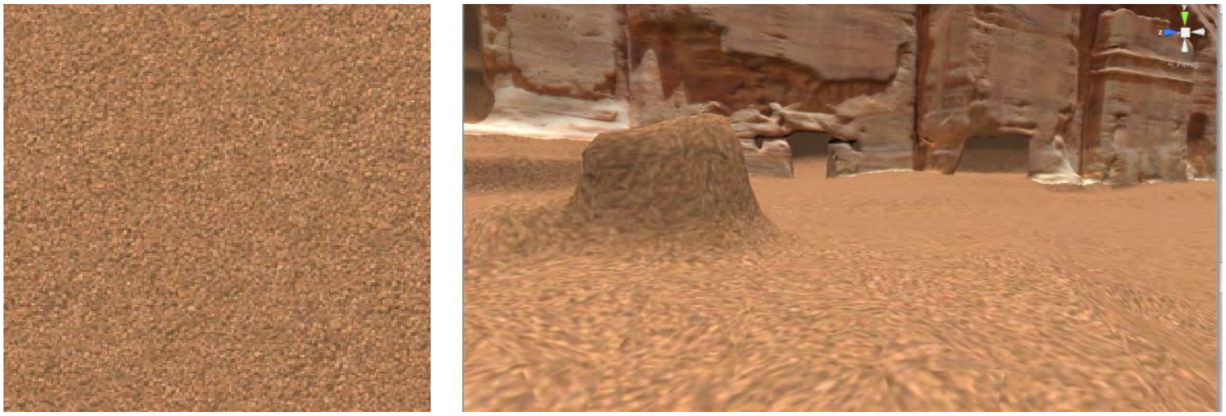


Figure 5.15: Left: Sample texture. Right: Sample texture applied to the ground in Petra.

5.3.4 Panoramas

Panoramas were taken in front of and inside of important monuments of Petra and in the landscape between monuments. Figure 5.16 shows the locations of all panoramas. For the purpose of this VW a selection of the most relevant panoramas was chosen to be included.

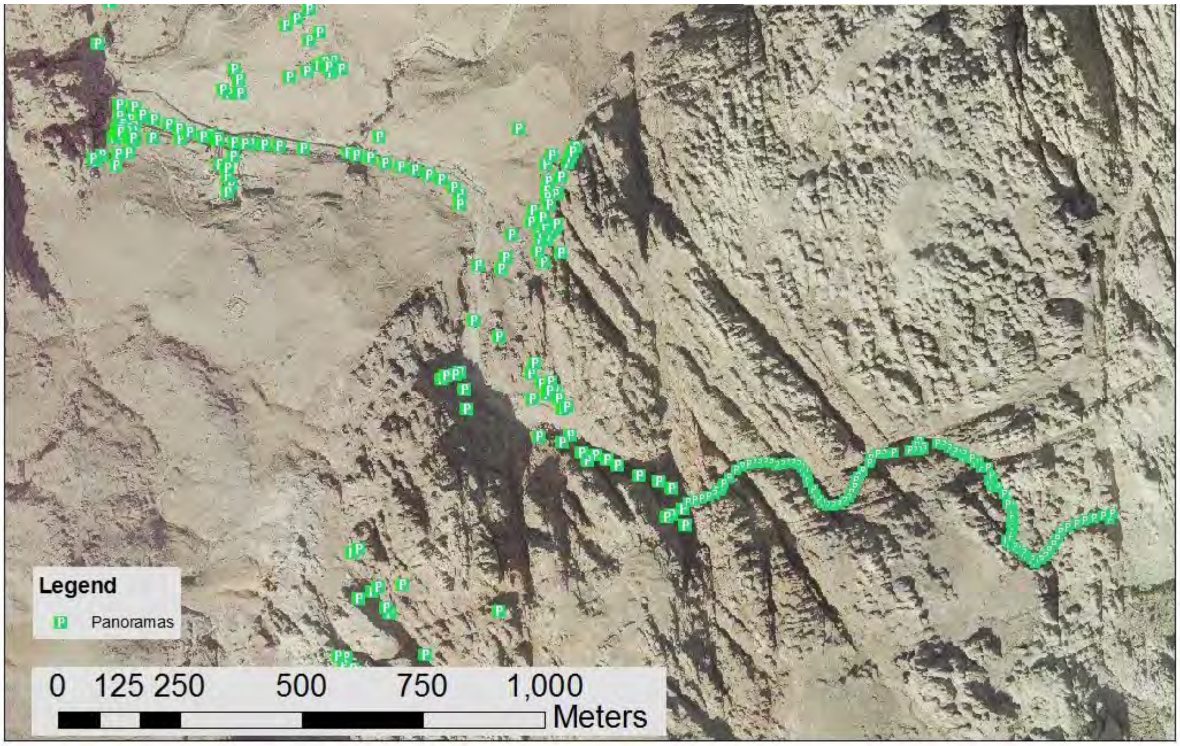


Figure 5.16: Panorama locations in Petra.

Panorama images were captured with a fish-eye lens using a panoramic tripod head that maintains the external pupil/perspective center of the camera to be in the same position. Six images are taken at 60 degrees horizontal interval and one taken upwards towards the sky. The images are later stitched together into one seamless image representing a 360 x 180 degree field of view.

Figure 5.17: Top: Seven individual panorama images taken with a fisheye lens. Bottom: The individual images are stitched together to form a 360-degree panorama image.

Where the panorama images were captured in areas of high lighting contrast, such as inside the dark rooms of monuments with windows letting in bright sunlight, bracketed images were recorded. These consist of a sequence of images of the same scene with different exposure settings. Longer-exposed images capture the dark areas of the scene and images taking with short exposure times capture the brightly lit up areas. The images are then blended together using the best lighting conditions from each of the sets of bracketed images. In this way a visually appealing panoramic image is stitched together. Refer to Appendix B.2.4 on how panoramas were imported into Unity.

5.3.5 Avatars

Avatars of historical figures with built in AI were not included in the VW as they are beyond the scope of this research. Extensive time and man-power would have been necessary to create realistic, historically accurate avatars. Their inclusion should be considered for future versions of the VW.

5.3.6 Flora

It was decided not to include any flora in the VW as hardly any flora exists in Petra except for a few small shrubs. Additionally, the artificial nature of flora detracts from the highly objective and realistic 3D models.

5.3.7 Virtual Objects

Virtual objects are outside the scope of this research, but could be incorporated in later versions of the VW.

5.3.8 Audio

Audio, in the form of music, ambient sounds and narration is beyond the limitations of this research, but could be incorporated in later versions of the VW.

5.4 Design elements

Creating and incorporating the design elements identified in Chapter 4 into the VW is discussed in the following section.

5.4.1 Splitting up the VW into scenes

To aid running speeds of user‘s computers it was decided to split the VW up into five separate scenes (see Appendix B.1.2 for a description of a scene), which represent five geographical areas, of Petra (see figure 5.18). In this way not all the 3D models of Petra need to be loaded at once. Reducing the number of polygons loaded into a scene (by assigning only a few 3D models per scene) reduces the start-up load time and running speed of each scene in the VW. The splitting of the VW was done according to geographic groupings of monuments. The following is a list of monuments in each scene.

Scene one	Scene two	Scene three	Scene four	Scene five
Exit of Siq	Amphitheatre	Qasr al Bint	Landscape of Wadi Farasa	Monastery
Area surrounding Treasury	Urn Tomb	Great Temple	Garden Tomb	
The Treasury	Corinthian Tomb	Temanos Gates	Soldiers Tomb	
Tomb Opposite the Treasury	Silk Tomb	Winged Lion Temple	Triclinium	
Passage from Treasury To Façade Tomb	Palace Tomb		Renaissance Tomb	
Façade Tombs				
Tombs Opposite Façade Tombs				
Tomb 825				
Tomb above 825				

Table 3: Splitting Petra into five separate scenes

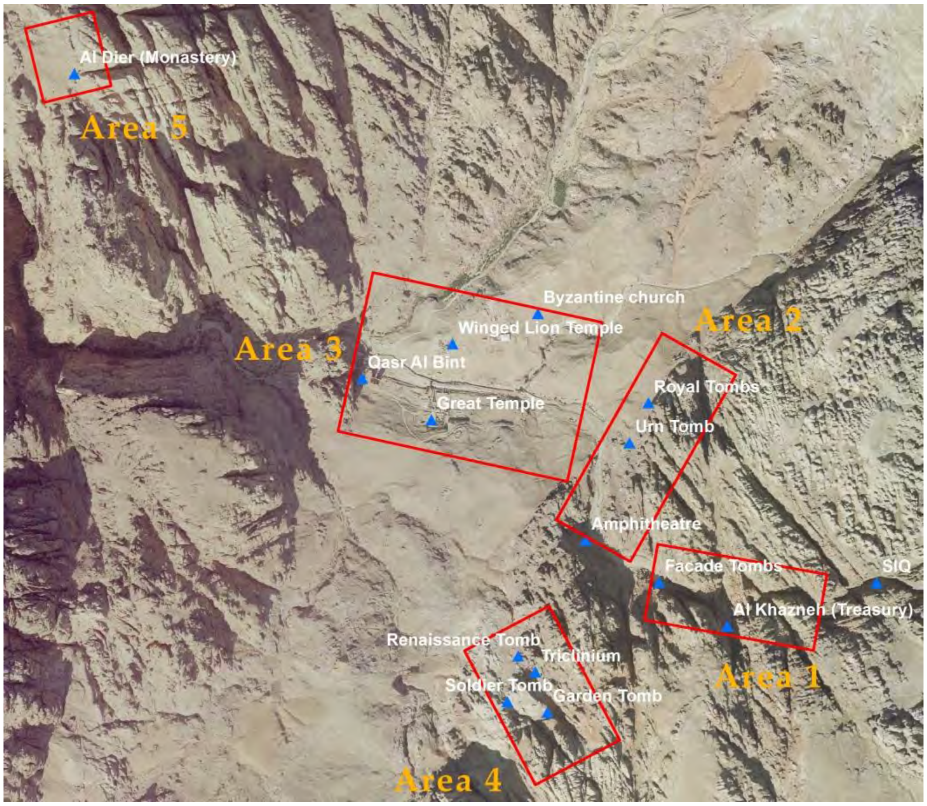


Figure 5.18: The chosen terrain of Petra was split into five scenes.

5.4.2 Start screen

The start screen is the first screen that the user is presented with when starting the VW. From this screen the user can choose which area (scene) of Petra to explore, although it is suggested to start exploring the VW in geographical order, from scene one through five.

Welcome to the Petra Virtual World!

Welcome to the Petra Virtual World!

Press "W" or up arrow to move forward
Click and hold mouse to change view direction
Press spacebar to jump
Click on Panorams to View
Use scroll wheel to zoom

Click on Mini Map to jump to location
Drag right mouse button to pan
Mouse scroll to zoom
Click on Panorama to View

Welcome to the Petra Virtual World!

The Petra 3D Virtual World was designed and created by:

Stephen Wessels
Zamani project ©
www.zamaniproject.org

In partnership with UNESCO ©

For more info contact:
stephen.wessels@zamaniproject.org

Figure 5.19: Start screen. Top: Area map with options on which area to load. Bottom left: Navigation information. Bottom right: Information about the creators and copyright holders of the VW.

The start screen was designed using the UnityGUI scripting functionality. A script was created and attached to a camera in the start scene which displays the start screen GUI, i.e. the buttons and image. The top-view image of Petra is designed to present the user with a geographical display of where the five different area scenes are located in Petra. Other menu options explain to the user how to navigate and give background information to the VW.

The first three buttons, ―Navigation”, ―Start”, and ―About”, on the start screen are toolbar buttons that, when clicked, call up different information in the main area of the window to be displayed. This can be seen in figure 5.19. The navigation toolbar button, when clicked, replaces the top-view image with text informing the user how to navigate in the VW. The ―About” toolbar button, brings up text that tells the user about the VW.

When the Start toolbar button is clicked and the top-view image is displayed the user has the option of which scene to load. Appendix B.3.1 gives further details on the creation of the start screen.

5.4.3 Virtual world screen layout

Once a scene has been loaded the VW of that scene is displayed. The screen layout of the VW was divided into three areas: The virtual environment (Area 1), the options-box (Area 2) and the mini-map (Area 3). The side length of the square display areas of Area 2 and 3 are defined as equal to half the screen height. This is so that on any screen the size of 1, 2 and 3 will adjust accordingly to the user‘s screens dimensions (see figure 5.20).

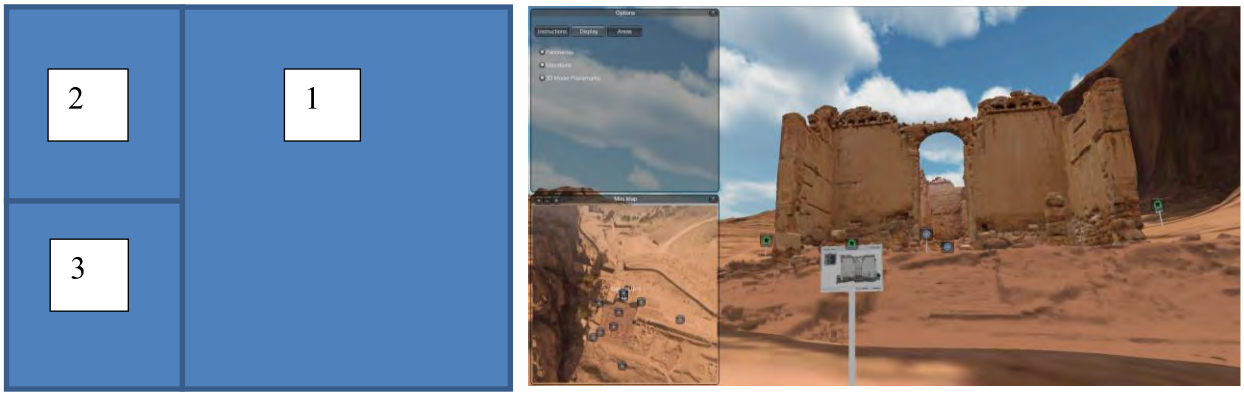


Figure 5.20: Left: Screen layout of the VW. Right: Screenshot of the VW of Petra.

Area 1 is dedicated to the 3D virtual environment. The visitor is placed in a first person perspective view here and is able to navigate through the virtual environment. Area 2 and 3 can be minimised so that area 1 is full screen.

Area 2 is used to display data and provide options and information about the VW. This includes information on navigation, how to use the minimap and options to turn on and off data in the main VW window and mini-map (Area 1 and 3). The user can also choose to load other areas of Petra here.

Area 3 is assigned to a mini-map of the entire landscape that the VW is covering, in this case the principal Wadis of Petra. The visitor will see the orthophoto of Petra and a green dot to show the location of the visitor in the middle of the displayed scene. When the visitor moves through the VW environment, the mini-map centre will follow. The mini-map may be zoomed or panned. The visitor can jump/teleport to any location on the mini-map by clicking on the desired location in the mini-map.

5.4.4 Area 1: Virtual environment

The terrain, orthophoto and 3D models were created and imported into their respected scenes in the VW as described in section 5.3 and Appendix B.2. A script, provided by Unity, called “mesh collider” was added to these objects which makes them virtually “solid” so that the user cannot pass through them.

5.4.4.1 Sky

To create a virtual sky the skybox functionality in Unity was used. As described in Appendix B.2.4 a skybox is a panoramic image, set at an infinite distance, acting as the backdrop to the scene, with all objects displayed in front of it. A standard skybox was chosen from the default skyboxes that form part of Unity.

Figure 5.21: Skybox example.

5.4.4.2 First person perspective view and First Person Controller

It was decided to use a first person view perspective in the VW. The reasons for this are three-fold. Firstly a first person view creates a more immersive experience when exploring the site. The virtual visitor sees the site as if through their own eyes. Secondly a third person perspective, where an avatar is visible in the scene and controlled by the virtual visitor, draws attention away from the objectives of the VW, which are the appreciation of the virtual environment itself. Thirdly, to create an avatar that is realistic and of good visual quality requires a large amount of research and development. This is outside the scope of this research.

Unity offers a ready-made first person controller (FPC) with first person viewing functionality. This FPC was modified and made use of in the VW as described in Appendix B.3.2

5.4.4.3 Navigation

To navigate in the VW the virtual visitor controls the movement of the FPC which in turn controls the view captured by the FPC‘s camera. This camera‘s view is then displayed on the user‘s screen. See Appendix B.1.4 for information on camera objects in Unity.

The standard navigation script controllers offered by Unity were found to be unsuitable for the objectives of this VW, which are to make navigation as simple as possible. It was therefore decided to modify the FPC controller scripts to better suit these objectives.

After some experimentation the following navigation controls were decided upon: Clicking and holding the left mouse button and moving the mouse will change the direction of view. Pressing the up arrow or the letter ―W” on the keyboard will move the user forward. Pressing ―spacebar” will cause the user to jump, which is necessary when climbing stairs or jumping onto higher objects or over walls.

Appendix B.3.3 describes in detail the modifications made to the navigation scripts to enable the functionality of the controls described above.

5.4.4.4 Additional pop-up information

To add functionality and information displays to the VW pop-up features were conceptualised and created. These features are in the form of panoramas, signboards and unique viewing platforms and exist as objects in the VW. Scripts were coded in Java and attached to each object to enable the desired functionality of each feature. The script places a hovering icon, with attached tooltip (descriptive text that appears when the mouse hovers over the icon), above the object in the virtual environment and when the virtual visitor clicks the icon the feature‘s functionality is activated.

5.4.4.4.1 Panoramas

The VW was designed to provide the visitor with the panorama tour functionality, as described in section 3.2, in addition to the free navigation through the virtual environment provided by the Unity concept. The panorama tour was thus integrated into the VW to be called up by the virtual visitor as required.

To incorporate the panorama tour functionality a panorama object was created in Unity in the form of a thin cylinder with a clickable icon, and associated tooltip, hovering over it. The panorama was then placed in the virtual environment at the exact location where the panorama was captured in Petra as can be seen in figure 5.22 below. The panorama placement icons also appear in the mini-map.

Figure 5.22: Left: Panorama positions, indicated with cylinder and blue icon, in the virtual environment. Right: Panorama positions on the mini-map, indicated by blue icon.

Each panorama was then aligned to the Unity co-ordinate system axis so that a virtual visitor on entering the panorama is placed into the same viewing orientation as in the VW scene. In this way jumping into and out of each panorama appears seamless.

Refer the Appendix B.3.4.1 for details on the creation of the panorama tour functionality

To exit the panorama the forward ―W” button or up arrow is pressed which switches the camera depth values in the camera hierarchy back to their original values so that the FPC camera once again has display priority over the panorama camera. The virtual visitor then walks forward out of the panorama view and into the virtual environment view provided by the FPC camera. Since the panorama view has been aligned to the virtual scene view the visitor view orientation of the FPC is maintained as can be seen in the figure 5.23 below.

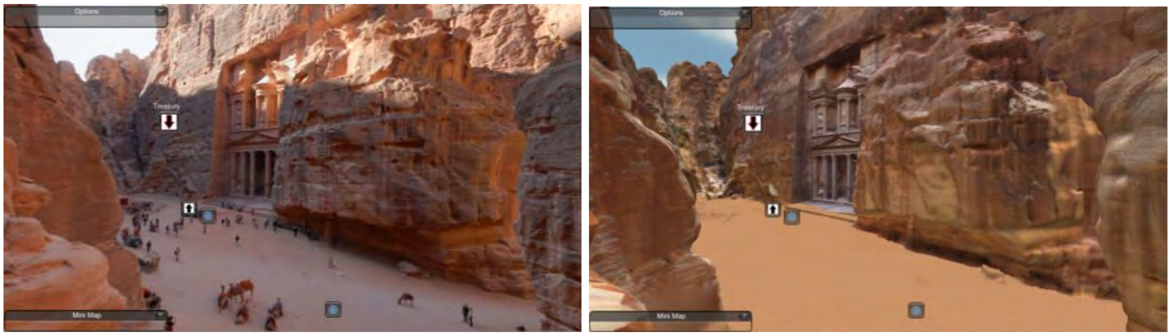


Figure 5.23: The two views of the VW from the same location in the virtual environment show the exact alignment of a panorama in the VW. Left: Panorama view. Right: Virtual environment view.

5.4.4.4.2 Elevations, sections and plans

Elevations, sections and plans, as described in section 4.1.3, were introduced into the VW as virtual signboards and pop-up images. In front of each monument in the VW a signboard object was positioned that displays elevations, sections, or plans of the monument. The signboard object consists of a thin cylinder with a rectangle plane attached to its top. On this plane the elevation, section or plan is placed, as can be seen in figure 5.24.

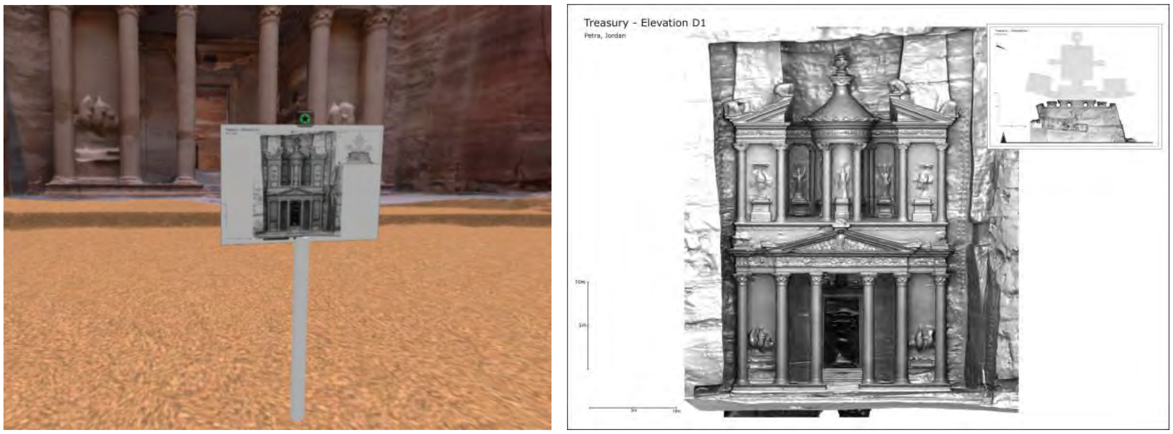


Figure 5.24: Left: Elevation signboard in the VW. Right: Elevation of the Treasury, Petra.

Hovering above the signboard is a green star icon, with associated tooltip. When the green star icon is clicked, a full screen display of the elevation, section or plan appears on screen. An exit button at the top of the screen removes the image. See Appendix B.3.4.2 for details on how the sections, elevations and plans pop-up functionality was created.

5.4.4.4.3 Raised viewing platforms/3D model placemarks

The visitors viewing perspective of the virtual environment and the 3D model monuments while exploring the VW is not limited to ground level as a raised viewing platform, in the form of an invisible ledge, was placed at an elevated height in front of each virtual monument. A hovering down-arrow clickable icon was then placed above these ledges to signal to the user that a ledge is available.

When this icon is clicked the user is transported to this invisible ledge. This offers the user a unique bird‘s eye view perspective of the monuments that would not even be possible with real site visits. The user can then walk off the ledge to explore the 3D model and surrounds. Each monuments viewing platform icon, with associated tooltips, also function as 3D model placemarks for the monuments in the virtual environment as they can be seen from any distance in the scene. They enable the visitor to quickly jump to monuments that they are interested to view. The placemark icons are also displayed in the mini-map, and can be clicked there to jump to the chosen monuments. See Appendix B.3.4.3 for details on how the raised viewing platform functionality was created.

Figure 5.25: Left: View of the 3D model of the Amphitheatre from ground level. Right: View of the 3D model of the Amphitheatre from the raised viewing platform.

5.4.5 Area 3: Mini-map

The mini-map serves to display the location of the virtual visitor in the virtual environment and indicate the geographical locations of the monuments and panorama positions in the nearby vicinity. The backdrop of the mini-map is the ortho-image or the textured 3D models. Functionality of the mini-map includes zooming, panning and teleporting, where the virtual visitor is transported to a location in the VW by clicking on that location in the mini-map.

The virtual visitor‘s location is displayed at the centre of the mini-map by a green icon and as the visitor moves through the scene the centre of the mini-map display follows accordingly. Clickable blue icons, with associated tooltips, on the mini-map indicate panorama positions in the virtual environment and when clicked transports the virtual visitor to the panoramas location and activates the panorama. Raised viewing platforms/3D model placemarks are indicated with a black and white down-arrow icon and when clicked transport the user to the location of the platform in the virtual environment. A single script was created to handle all the above functions, the details of which are described in Appendix B.3.5

Figure 5.26: The mini-map in the Petra VW.

5.4.6 Area 2: Options-box

The options box was designed to display information about the VW and provide users with options affecting what is displayed in the VW scene. The options box can be minimised in the same way that the mini-map is minimised (by clicking the minimising icon) so that Area 1, the 3D virtual environment, can be displayed in full-screen mode.

The code that draws the options box window is part of the mini-map script that is attached to the mini-map camera. The first task of the options-box portion of the script is to create a semi-transparent box, with the text ―Options" as its label, to display in the top left corner of the screen. Three toggle buttons, ―Instructions", ―Display" and ―Areas" are then displayed in this box. When a toggle button is clicked, it shows a greyed out background, and its functionality is displayed. Each toggle button‘s functionality is described in the following text. Figure 5.37 shows the layout of the options box display.

Press "W" or up arrow to move forward
Click and hold mouse to change view direction
Press spacebar to jump
Click on Panoramas to View
Use scroll wheel to zoom

Click on Mini Map to jump to location
Drag right mouse button to pan
Mouse scroll to zoom
Click on Panorama to View

● Panoramas
● Elevations
● 3D Model Placemarks

Load Area 1	Treasury, Facade Tombs
Load Area 2	Royal Tombs, Amphitheatre
Load Area 3	Qasr al Bint, Great Temple
Load Area 4	Wadi Farasa
Load Area 5	Monastery

Figure 5.27: Options-box display in the VW. Left: Instructions. Middle: Display. Right: Areas.

5.4.6.1 Instructions toggle button

The instructions toggle button (figure 5.27) displays text that describes to the virtual visitor how to navigate in the VW.

5.4.6.2 Display toggle button

The display toggle button (figure 5.27) provides the user with a choice of what pop-up information to display in the virtual environment. The three options, in the form of checkboxes, are for panoramas, elevations, and 3D model placemarks, which are displayed in the virtual environment as floating icons, in the form of blue dots, green stars and down-arrows respectively. Unclicking the checkbox removes the icons from the scene. In the case of the panorama location icons and the 3D model placemarks, which are also displayed on the mini-map, the checkbox removes the icons from both the virtual environment and the mini-map. These options exist so that the scene can be viewed as if the user was there in real life without artificial floating icons cluttering the screen.

5.4.6.3 Areas

The areas toggle button displays a choice of which alternate area scene the user wishes to explore. Also the start screen can be loaded from here. See figure 5.27.

5.4.7 Lighting

A combination of ambient light and directional light was introduced to provide lighting in the VW. Ambient light can be modified in the Unity rendering settings and can be increased or decreased in intensity. Through investigation it was found that the ambient light can only be increased to a certain level before the entire scene appears over-exposed. A value exactly halfway between complete darkness and complete brightness was thus chosen to provide light for the entire VW.

Figure 5.28: Ambient light in the VW. Left: Underexposed appearance. Middle: Overexposed appearance. Right: Midway exposed ambient light provides the most natural appearance.

To provide realistic lighting in the VW a directional light was chosen to emulate the sun. Terrain and objects are lit up by directional lights to provide a natural appearance of the scene that includes shadows cast onto the scene by the 3D objects. It was found that using one directional light was not sufficient to light up the entire virtual environment as places with very high narrow walls allowed insufficient light to reach the ground level and so certain areas, especially those cast in shadow, appeared very dark. Initially, to overcome this problem, a torch was attached to the FPC that illuminates the scene directly in line with the visitor‘s viewing direction. This was done by attaching a spotlight to the FPC facing the same direction as the FPC faced. This torch then lit up a circular area in front of the visitor. After testing this solution in the VW it was found that the torch light appeared unrealistic, as the visitor would expect a real-life scene to be lit up in its entirety and not in a limited area as is the case when a torch is used.

The second attempt to overcome the dark areas problem was to direct the directional light to always be oriented in the same direction in which the visitor is facing. As the visitor turns in the VW the directional light also turns to follow this movement. This approach was found to be better than the torch solution, but still unsatisfactory. When the visitor is looking forward the areas to the left and right of the scene appeared darker, but as the user turned to look at these dark areas they would magically light up. After testing this approach this solution was found to be unrealistic and it also created a feeling of always wanting to turn to see what was waiting in the dark.

Finally a solution was found whereby five directional lights were used. This cluster of lights was placed above the centre of the scene and each light was rotated to face at different directions. The directions chosen were roughly 70 degrees apart in the horizontal axis (360/5). Each light pointed downwards 50 degrees. This approach was found to be the most suitable as equal light was dispersed throughout the scene despite the high walled canyon areas.

Chapter 6

Summary of activities in the creation of the VW

The points below summarise the input data and design elements and functionality that were created for the Petra VW.

6.1 Input data

- **Terrain**
    - A 1m grid of terrain data was acquired from aerial imagery, a portion of which was used as the terrain in the VW, covering an area of 1.21 square kilometres (1.1km x 1.1km).
    - A 3km stretch of terrestrial laser scan data was registered together and individual scans were cleaned and modelled to form a 0.35 square kilometre landscape model.
    - These two terrain data sources were merged together to create the most accurate and detailed model of the terrain as possible. This model was used as the terrain of the VW, by converting it into a format suitable for Unity.

- **Ortho-image**
    - An otho-image was created from aerial imagery and draped over the terrain to create an image covering 1.21 square kilometres that was used as a texture for the terrain in the VW. The pixel resolution of this texture was 0.3m.

- **Textured 3D models of monuments**
    - 24 3D models were created and most of them textured. To create the models the individual scans had to be cleaned, registered and hole-free modelled. They then had to be decimated and textured using images taken on site.

- **Panoramas**
    - The most relevant panoramas, which were captured in front of important monuments and at good vantage points in the terrain, were chosen to be included in the VW. These 360-degree panoramas were stitched, enhanced and added to the VW with panorama tour functionality.

6.2 Design elements and functionality

- **Splitting up the VW**
    - The VW was split into five separate scenes to aid loading and running performance.

- **Start screen**
    - A start screen GUI was created that includes a map to show the location of the five areas, information on how to navigate, and information on copyright of the VW.

- **Virtual environment**

  - The main screen area of the VW shows the first person perspective view of the virtual environment.
  - A FPC character was added which the user controls and the virtual environment is seen through a camera attached to this character.
  - The scripts controlling the character were modified to be more user-friendly
  - Artificial sky was added in the form of a skybox.
  - A script was written to display clickable icons where the panoramas were placed in the virtual environment. The script transports the FPC to the location of the panorama and displays the panorama when clicked.
  - A script was written to display clickable icons where the Elevations, Sections or Plans were placed in the virtual environment. The script displays a full screen view of the image when clicked.
  - A script was written to display clickable icons where raised viewing platforms are placed in the virtual environment. The script transports the FPC to the location of the platforms so that the user can experience unique views of the 3D model monuments.

- **Mini-map**

  - A script was written to draw a scalable mini-map window in the bottom left corner of the screen depending on the user‘s screens dimensions. The script includes functionality to allow for:
    - The centre of the mini-map to always follows the FPC.
    - The mini-map to be zoomed and panned using the mouse and mini-map buttons.

    - > The mini-amp to be minimised by clicking on the minimisation icon in the map window.
    - > Teleporting – the FPC character is transported to a clicked position on the mini-map.
    - > Panoramas and raised viewing platforms appear in the mini-map as clickable icons. Clicking on them activates their functionality.

- **Options box**

  - A script was written to draw a GUI options-box window in the top left corner of the screen. The options box includes three toggle buttons:
    - > Instructions: Provide instructions on how to navigate in the VW.
    - > Display: Turn panoramas, 3D model placemarks and pop-up info icons on or off in the virtual environment display area.
    - > Areas: Options to load alternate area scenes.

- **Lighting**

  - Lighting was achieved by setting the ambient light in the scene and creating five directional lights shining from a central position onto the scene.

Chapter 7

Discussion

7.1 Discussion on the creation of the VW

In chapter 3 it was argued that the use of a game engine for the creation of a virtual heritage environment is highly desirable as it allows the developer access to a full software development kit to create a VW without having to first create software for this. However, using a game engine to build a virtual environment should not be considered trivial as the game engine supplies only the basic functions that virtual environments share. The game engine serves primarily to real-time render the virtual environment as the user explores a scene. All the content must be created by the developer and additionally all the functionality in the virtual environment must be coded via scripts that are then attached to the created content. The online Unity forum provided many answers to questions on coding, thus assisting the author, who is not an experienced programmer.

Unity was found to be highly suitable for the task of creating a VW of the heritage site of Petra. Being able to run the VW in Unity at any stage via the ‗play' button, as described in Appendix B.1, helped the development of the VW succeed in an iterative fashion. For example the brightness of the directional lights could be adjusted and quickly tested in real-time by running the VW to see the results instantly. Most of the input data and design elements as described in chapter 4 could be implemented in Unity which proves the suitability of Unity for the creation of VWs.

7.1.1 Input data

The input data consisting of terrain and orthophoto from aerial images, textured 3D models from laser scanning and imagery, and panoramas all had to be created for their use in the virtual environment in Unity in mind.

Unity imposes limitations on the input of data in terms of the file size of models and images. As described in section 5.3.2 an image imported into Unity can have a maximum resolution of 4096 x 4096 pixels and a 3D model can be no bigger than 65000 polygons. These limitations had implications on the input of data.

An easy method of overcoming the 3D model size limitation is splitting a model up into blocks, which Unity does automatically if the imported model is bigger than the stipulated size. However, the terrain could not be split up since the method of importing the data through a plug-in tool that converts a polygon model into a heightmap, only works for models of up to 65000 polygons. This forced a compromise between size of the terrain area to be incorporated in the environment and amount of detail that is made available. The end result was a terrain of roughly 4m resolution which was deemed to be somewhat lacking in detail. The orthophoto, which was draped over the terrain, had a limit of 4096 x 4096 pixels which resulted in a 0.3m pixel resolution. This resolution lead to a visually satisfactory texture and masked the effect of the low resolution terrain model.

The 3D models of the monuments of Petra had to be created in such a way that no erroneous polygons were present and no holes existed in the models. Hole-filling is a concern in heritage site documentation as the authenticity of the models is compromised when holes are artificially filled. It was necessary to fill holes in this case as the appearance of a hole in the model would have distracted the user from experiencing an immersive, lifelike virtual environment. The 3D models had to thus undergo manual, time-intensive cleaning and hole-filling. Additionally, the models had to be heavily decimated to minimum polygon counts which still displayed relevant and realistic detail. Exploring texturing methods and subsequent texturing of the 3D models resulted in a new approach to texturing by the Zamani project. In this approach high quality texture maps were created from a minimum number of images taken on site that were used to texture low resolution 3D models. Additionaly, two or more texture maps were created to texture 3D models that had a large amount of surface area to be textured. A continuous texture was created from the Siq to tomb 825 which stretches over an extent of 500m. This was achieved with extensive manual image alignment, masking of images, and image colour manipulations of the approximately 1500 images used for this area of terrain. Also the models had to be broken up into many parts that were then textured individually at different resolutions depending on their potential proximity to the user in the virtual environment. The final textured models were found to be visually very appealing.

The chosen panoramas in the tour were carefully stitched together so as to avoid any stitching artefacts, blurry merging of edges of individual panorama images, and provide consistent colour. All Panoramas had to be aligned to the co-ordinate axes of Unity which proved to be challenging, especially where there were no close-by features for reference. The addition of the panoramas added greatly to the visual experience of exploring the VW by allowing a user to have a completely realistic view of Petra from certain positions on the terrain. The authenticity of a 360-degree panorama cannot be questioned and this functionality thus provides for a realistic impression of Petra even if the terrain or 3D models surrounding the panorama position are not perfect. By allowing the user to _walk' out of the panorama into the virtual environment and having the general scene remain the same (due to the panorama being aligned to the scene) the transition appears seamless. A user does not get this seamless experience when jumping from panorama to panorama in a panorama tour.

Avatars, flora and virtual objects were not included in the VW of Petra as they fall outside the scope of this project. However these elements are desirable in a VW as discussed in Chapter 4.1. Avatars can introduce meaningful intangible heritage into the VW by way of historical figures and narration. Flora adds a sense of presence and realism to a virtual environment especially when it is made to move in an artificial breeze. Virtual objects, besides being tangible heritage data, can also introduce intangible culture when meaning and historic stories are attached to these objects. Ideally all three elements should be included in a VW to add intangible heritage to the virtual environment and the VW of Petra suffers without them.

7.1.2 Design elements

By splitting the VW into five separate scenes the initial load times of the VW was decreased and the navigation though the virtual environment ran smoothly. The unfortunate consequence of this decision was that it takes some time to switch between scenes as the load up time of each scene is a few minutes. This waiting for scenes to load disturbs the flow of the VW experience.

The start screen, created using the UnityGUI scripting commands, is functional but could be visually more appealing. With more time invested the load screen could be improved by making the map, showing the layout of Petra with the locations of the different scenes, clickable.

The VW screen layout was designed with user practicality in mind with the screen dimensions of the virtual environment, mini-map and options-box being automatically adjusted to users‘ screens. The options-box could be made more appealing my turning the buttons into pull-down menu options.

Creating scripts to provide functionality in the VW was found to be challenging, but manageable. The many tutorials offered online and the community forums helped immensely in writing the scripts. The functionality that was created can be seen as proof that a VW can be created by a developer with limited programming experience.

7.2 Navigation in the VW

The techniques used to navigate in the VW (via keyboard and mouse) are the same techniques used in the highest quality video games. These techniques have thus been tested thoroughly by their implementation in these games which are designed with the premise that the best games are the most simple to use, but difficult to master (Svånå, 2010). Using these techniques in the field of virtual heritage may be new but the techniques themselves are not. Their value is established regardless of the environment they are used in.

The ability of the user to navigate through the VW in an efficient manner is largely dependent on the user‘s hand-eye co-ordination and skill level in operating keyboard and mouse while moving through the environment. Ideally training should be offered to each user to teach the navigation controls but this is not practical as the VW is meant for a world-wide audience. Some users will thus struggle to navigate in the VW without the suitable skillset needed to operate the controls. Basic instructions are offered to users on how to navigate, but these may be insufficient. Instructional videos or tutorials on navigation through the VW should therefor complement the basic instructions.

Chapter 8

Conclusions

The principal objective of this research was to create a virtual world of Petra using the spatial data of Petra captured and produced by the Zamani project in co-operation with UNESCO, ISPRA and Jordanian partners. The VW is intended for use by a world-wide audience consisting of any interested parties that wishes to explore Petra for any reason. The benefits are in the areas of education, planning and site management. The VW can be considered a success and is available online and viewable in a web-browser after installing the free Unity plug-in.

The field of virtual heritage is growing and becoming increasingly prominent in the public‘s eye as new technology, equipment and software make it possible to document heritage sites with greater detail and efficiency and display this data using visually appealing methods. Thus far gathered documentation has not been widely used and its potential is not yet fully exploited (Ruther et al., 2012). Presenting this data has been a challenge that virtual heritage practitioners have to address. Methods of presenting to the data to the general public on an easy to use platform have led to the embracing of techniques from other industries, for example game engines.

From the case studies reviewed in chapter 3 it was found that many virtual environments of heritage sites exist, with many of them being created with game engines. However, all these virtual worlds were based on reconstructed 3D models of heritage structures - that often no longer exist - using primitives and sample texture as opposed to 3D models created from data recorded of the physical remains of a heritage site. This VW might be the first to employ textured 3D models and 360-degree panoramas of digitally documented heritage sites.

The first research question explores the integration of spatial data of a heritage site into a seamless virtual environment. The VW of Petra is presented as proof of concept of a seamless VW created from the digital documentation of heritage sites by using the Unity game engine.

Spatial data was either imported directly into the virtual environment in the form of terrain and 3D models or included in the VW by way of added functionality, which included panoramas, sections, elevations and plans. Using the VW functionality created in this research input data of other documented sites can be used to develop VWs of other heritage sites.

The second research question asks how textured 3D models and landscapes can be created to be incorporated into the VW bearing in mind computer processing power. Models were hole-filled and intelligently decimated and a novel texturing method was employed to create good visual appearance of low resolution models.

The third research question examines how 360-degree panoramas can be included in a 3D virtual environment. This was achieved by placing the panoramas in their exact locations in the virtual environmental and aligning them to the scene.

The fourth research question investigates the creation of a VW characterised by ease of use. The VW is easily accessible as it is available on the Zamani project website (Zamani, 2015). It is easy to navigate as it uses standard navigation techniques employed by popular video games. The functionality has been designed to be as intuitive as possible with clickable icons hovering in the environment which activate functionality when clicked.

The fifth research question is the most difficult to answer as it attempts to establish how the VW can be designed to be an engaging and immersive experience where the virtual visitor is eager to explore. The heritage site of Petra is globally acknowledged as one of the most spectacular sites in the world to visit and so the VW was created with the objective of recreating the physical nature of the site to be as visually appealing as possible and as easy to use as possible. To achieve this the VW was populated with high quality spatial data products that provide an environment with a realistic appearance. The textured models have a high visual appeal which it is hoped will entice the visitor to explore further and remain captivated by the site.

With the design and creation of virtual worlds of heritage sites as described in this thesis we are a step closer to the prediction of DeLeon and Berry (2000) who envisage an online repository of virtual heritage site environments that can be accessed from anywhere in the world. The benefits of this include the creation of awareness of cultural heritage sites, and the creation of a platform for the subsequent education, site management, analysis and other fields of research that can be associated with virtual heritage.

8.1 Recommendations for future research

The Petra VW can be improved by using higher quality models and textures, and by including more monuments. Lighting of the VW should be further explored.

Visualising the VW is currently by way of desktop computers. The VW should be incorporated into Virtual Theatres, HMDs and CAVES which would enhance the immersion of the user and the experience of exploring the Petra virtual environment.

This VW can be expanded in a number of different ways to target more specific audiences. The creation of an educational game in Petra, virtual tour guides and multiplayer functionality are some of the options for this expansion. Additionally the VW could be made into a site management tool where a user could make virtual annotations while navigating through the site and view the site from unique vantage points.

AI bot characters and animals could also be added to the VW as well as audio in the form of traditional songs. These elements would add intangible content to the VW which would enhance the user‘s experience of virtually visiting the site.

8.2 Outlook

VWs are likely to improve significantly in the future as technology advances. Virtual heritage needs to embrace these advancements to the benefit of virtual heritage applications. These advancements include hardware capabilities, software, and methods of displaying the virtual heritage data. Advancements in the recording of sites will also be to the benefit of virtual worlds. Technology such as drones can be used to capture data from highly beneficial viewpoints. Scanning from mobile platforms is already available and this can be applied to recording heritage sites in much shorter time frames than terrestrial laser scanning. The use of SfM software for the recreation of structures using standard cameras is offering an easy and affordable method of recording heritage sites. The film and computer game industries are continuing to grow and VH can and must make use of the advancements in these fields.

References

Ahmad, Y. (2006) ‗The Scope and Definitions of Heritage: From Tangible to Intangible‘, *International Journal of Heritage Studies* Vol. 12, No. 3, May 2006, pp. 292–300.

Aziz, K. A. and Siang, T. G. (2012) 'Virtual Reality and Augmented Reality Combination as a Holistic Application for Heritage Preservation in the UNESCO World Heritage Site of Melaka', *Change* 25(1.30), 60—6.

Bailey, K. (2015) *Unity 5 versus Unreal Engine 4, and What It Means for You.* Available at: http://www.usgamer.net/articles/gdc-2015-unity-5-versus-unreal-engine-4-and-what-it-means-for-you (Accessed: 20 May 2015).

Bandara, K.R.M.U., Samarakoon, L., Shrestha, R.P., Kamiya, Y. (2011) ‗Automated Generation of Digital Terrain Model using Point Clouds of Digital Surface Model in Forest Area‘. *Remote Sensing.* 3 845-858.

Bellotti, F., Berta, R., De Gloria, A., D'ursi, A., Fiore, V. (2012) 'A serious game model for cultural heritage', *Journal on Computing and Cultural Heritage (JOCCH)* 5(4), 17.

Brodkin, J. (2013) *How Unity3D Became a Game-Development Beast*, Slashdot. Available at: http://slashdot.org/topic/cloud/how-unity3d-become-a-game-development-beast (Accessed: 25 March 2014).

Ranzuglia, G., Callieri, M., Dellepiane, M., Cignoni, P., Scopigno, R. (2012) ‗MeshLab as a complete tool for the integration of photos and color with high resolution 3D geometry data‘, *CAA 2012 Conference Proceedings.*

Cawood, S., Fiala, M., Steinberg, D. H. (2007) *Augmented reality: a practical guide*, Pragmatic Bookshelf Raleigh, NC.

Champion, E. (2002) ‗Cultural engagement in virtual heritage environments with inbuilt interactive evaluation mechanisms‘, in *Proceedings of the Fifth Annual International Workshop PRESENCE,* 2002, pp. 117—128.

Champion, E. (2008), ‗Otherness of Place: Game-based interaction and Learning‘ in *Virtual Heritage Projects, International Journal of Heritage Studies*, vol. 14 no. 3, pp.210-228. Available at: http://www.tandfonline.com/doi/abs/10.1080/13527250801953686 (Accessed: 1 March 2014).

Champion, E. (2014) ‗History and Heritage in Virtual Worlds‘, In *The Oxford Handbook of Virtuality*, ed. Mark Grimshaw, UK: Oxford University Press.

Cruz-Neira, C., Sandin, D., DeFanti, T. A. (1993) *Surround-Screen Projection Based Virtual Reality: The Design and Implementation of the CAVE*. Paper presented at the SIGGRAPH '93.

DeLeon. V., Berry Jr, R. (2000) 'Bringing VR to the desktop: are you game? '*MultiMedia, IEEE* **7**(2), 68—72.

Dylla, K., Frischer, B., Mueller, P., Ulmer, A., Haegler, S. (2008) 'Rome reborn 2.0: A case study of virtual city reconstruction using procedural modeling techniques', *Computer Graphics World* 16, 25.

El-Hakim, S. F., Beraldin, J., A. Picard, M., Godin, G. (2004), 'Detailed 3D reconstruction of large-scale heritage sites with integrated techniques', *Computer Graphics and Applications, IEEE* 24(3), 21—29.

Federation of America Scientists (FAS). (2008) *Discover Babylon*, FAS. Available at: http://www.discoverbabylon.org/ (Accessed: 24 January 2015).

Foni, A. E., Papagiannakis, G., Magnenat-Thalmann, N. (2010) 'A taxonomy of visualization strategies for cultural heritage applications', *Journal on Computing and Cultural Heritage (JOCCH)* 3(1), 1.

Foundation of the Hellenic World. (2014) *Foundation of the Hellenic World*, Athens, Greece. Available at: http://www.fhw.gr/fhw/ (Accessed: 25 March 2014).

Frischerconsulting. (2013) *Romereborn*. Available at: http://romereborn.frischerconsulting.com/about.php (Accessed at: 1 November 2014).

Frischer, B., Fillwalk, J. (2012) ‗The Digital Hadrian's Villa Project: Using virtual worlds to control suspected solar alignments', in *Virtual Systems and Multimedia (VSMM)*, 2012, 18th International Conference, pp. 49—55.

Granström, H. (2013) *Elements in Games for Virtual Heritage Applications*. Master's Thesis. University of Skovde.

Guttentag, D. A. (2010) 'Virtual reality: Applications and implications for tourism', *Tourism Management* 31(5), 637—651.

Haines, E. (2012) Unity3D Community. Available at: http://wiki.unity3d.com/index.php?title=Object2Terrain (Accessed: 25 March 2014).

Harman, J., Wernecke, J. (1996) *The VRML 2.0 Handbook: Building Moving Worlds on the Web*. Reading, MA: Addison-Wesley Professional

Held, C. (2012) *Creating 3D Models of Cultural Heritage Sites with Terrestrial Laser Scanning and 3D Imaging*. Master's Thesis. University of Cape Town

ICOMOS. (2002), *International Cultural Tourism Charter,* Managing Tourism at Places of Heritage Significance. Adopted by ICOMOS at the 12th General Assembly Mexico, October 1999. Available at: http://www.international.icomos.org/charters/tourism_e.pdf (Accessed: 23 January 2015).

ICOMOS. (2007) *The ICOMOS Charter for the Interpretation and Presentation of Cultural Heritage Sites*, Edited by Ename. Oudenaarde Ename, 2007: 1–14. Available at: http://www.enamecharter.org/ (Accessed: 10 March 2014).

Jacobson, J., Holden, L. (2005) ‗The virtual Egyptian temple‘, in *World Conference on Educational Multimedia, Hypermedia and Telecommunication'*, pp. 4531—4536.

Jacobson, J., Vadnal, J. (2005) ‗The Virtual Pompeii Project‘, in *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, pp. 1644--1649.

Jacobson, J., Handron, K., Holden, L. (2009) 'Narrative and content combine in a learning game for virtual heritage', *Distance Education* 9(2), 7—26.

Jacobson, J., Sanders, D. (2013) ‗The Vari House; Digital Puppeteering for History Education‘ in *Journal of Immersive Education*, 2013, Institute of Immersive Education. Available at: http://publicvr.org/publications/Jacobson2013.pdf. (Accessed: 24 January 2015)

Johnson, C. (2005) *Virtual Tours of Dudley Castle*. Available at: http://www.exrenda.net/dudley/index.htm (Accessed: 24 January 2015).

Koutsoudis, A. (2012) *Bringing cultural heritage 3D content to the web using the X3DOM framework*. Available at: http://www.carare.eu/dut/Media/Files/Bringing-3D-Content-to-the-Web-Using-X3DOM (accessed: 24 January 2015).

Learningsites. (2014) *Learningsites*. Available at: http://www.learningsites.com/ (Accessed: 24 January 2015).

Lucey-Roper. M. (2006) ‗Discover Babylon: Creating A Vivid User Experience By Exploiting Features Of Video Games And Uniting Museum And Library Collections‘, in J. Trant and D. Bearman (eds.). *Museums and the Web 2006: Proceedings, Toronto: Archives & Museum Informatics.*

Mayden, A. (2015) Unreal Engine 4 vs. Unity: Which Game Engine Is Best for You? Available at: http://blog.digitaltutors.com/unreal-engine-4-vs-unity-game-engine-best/ (Accessed: 20 May 2015).

Musawwarat Graffiti Archive. (2011) Max Planck Institute for the History of Science and Humboldt University Berlin. Available at: http://musawwaratgraffiti.mpiwg-berlin.mpg.de/index_html (Accessed: 25 March 2014).

Noh, Z., Sunar, M. S., Pan, Z. (2009) ‗A review on augmented reality for virtual heritage system 'Learning by Playing'. *Game-based Education System Design and Development*, Springer, pp. 50—61.

O' Flanagan, J. (2015) Game *Engine Analysis and Comparison* Available at: http://www.gamesparks.com/blog/game-engine-analysis-and-comparison/ (Accessed:20 May 2015).

Osman, A., Wahab, N.A., Ismail, M.H. (2009), ‗Development and Evaluation of an Interactive 360 defree Virtual Tour for Tourist Destinations'. *Journal of Information Technology Impact. Vol. 9, No. 3, pp. 173-182, 2009*

Ott, M., Pozzi, F. (2010) ‗Towards a new era for Cultural Heritage Education: Discussing the role of ICT' in *Computers in Human Behaviour*, Vol 27, Issue 4, July 2011, pp 1365 – 1371.

Patel, U.K., Patel. P., Hexmoor, H., Carver. N. (2012) ‗Improving Behaviour of Computer Game Bots Using Fictitious Play'. *International Journal of Automation and Computing.* April 2012 122-134.

Petty, A. (2007) *Discovering Babylon: The Opportunities, Challenges and Irresistible Potential of Video Games as an Educational Medium*. SBL Forum. Available at: http://www.sbl-site.org/publications/article.aspx?articleid=672 (Accessed: 24 January 2015).

PublicVR, (2008) *PublicVR.* Available at: http://publicvr.org/ (Accessed: 24 January 2015)

Pujol, L. (2004) 'Archaeology, museums and virtual reality', *Revista digital de humanidades, UOC.* Available at: http://www.Uoc.edu/humfil/articles/eng/pujol0304/pujol0304.pdf (Accessed: 24 January 2015).

Roussou, M. (2002) ‗Virtual Heritage: From the Research Lab to the Broad Public', in F, Niccolucci, Virtual Archaeology: Proceedings of the VAST 2000 Euroconference, Arezza, Italy, 24-25 November 2000, p.93-100, BAR International Series 1075, Oxford, Archaeopress, 2002.

Rüther, H., Held, C., Bhurtha, R., Schröder, R., Wessels, S. (2011), ‗Challenges in Heritage Documentation with Terrestrial Laser Scanning', *Proceedings of AfricaGeo*, 2011

Rüther, H., Held, C., Schröder, R., Bhurtha, R., Wessels, S, (2012) ‗From Point Cloud to Textured Model, the Zamani Laser Scanning Pipeline in Heritage Documentation', *South African Journal of Geomatics*, Vol. 1, No. 1.

Rüther, H., Bhurtha, R., Wessels, S., Schröder, R. (2014) ‗Spatial Documentation of the Petra World Heritage Site', in *Africa Geo*, 1 – 3 July 2014.

Sanders, D. H. (2001) ‗Persuade or Perish: Moving Virtual Heritage beyond Pretty Pictures of the Past' in *Proceedings of the Seventh International Conference on Virtual Systems and Multimedia*, p.236-245. Available at: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=969676 (Accessed: 1 March 2014).

Sanders, D. H. (2014) 'Virtual Heritage: Researching and Visualizing the Past in 3D', *Journal of Eastern Mediterranean Archaeology and Heritage Studies* 2(1), 30—47.

Smithsonian Institute. (2014) Available at: http://3d.si.edu/ (Accessed: 27 March 2014).

Stanco, F., Tanasi, D. (2011) ‗Chapter 5: Experiencing the Past: Computer Graphics in Archaeology', in G. Gallo, V., La Rosa, F., Stanco, D., Tanasi, in *Radamante al computer. Archeologia e informatica nel mondo minoico: l'esperienza catanese, Regione Siciliana.*

Steuer, J. (1992) 'Defining virtual reality: Dimensions determining telepresence', *Journal of communication* 42(4), 73—93.

Stone, R., Ojika, T. (2000) 'Virtual heritage: what next?', *MultiMedia, IEEE* **7**(2), 73—74.

Svånå, D. (2010) *Environment re-creation methods for virtual heritage using a game engine with discernment of visual learning cues.* Master's Thesis. Norwegian University of Science and Technology.

TARA, Trust for African Rock Art, Nairobi, Kenya. Available at: http://africanrockart.org (Accessed: 25 March 2014).

UNESCO. (2003) *Convention for the Safeguarding of the Intangible Cultural Heritage*. 32nd Session of the General Conference, Paris, 29 September – 17 October 2003.

Unity 3D 2014, Unity 3D, San Francisco, USA. Available at: https://unity3d.com/company/public-relations (Accessed: 25 March 2014).

Unity Vs Unreal Engine 4: Which Game Engine Comes Out On Top? (2015). Available at: http://www.teksmobile.com.au/blog/unity-vs-unreal-engine-4-game-engine-comes-top/ (Accessed: 20 May 2015).

Unreal Engine. (2015) Cary, North Carolina, USA, Epic Games. Available at: www.unrealengine.com (Accesssed 20 May 2015).

Venice Charter. (1964) ‗International charter for the conservation and restoration of monuments and sites', in *IInd International Congress of Architects and Technicians of Historic Monuments, Venice.*

Web3D Consortium. (2012), *What is X3D.* Available at: http://www.web3d.org/sites/default/files/page/About%20Web3D%20Consortium/What_is_X3D_201206.pdf (Accessed: January 2015)

Weis, A., Jacobson, J., Darnell, M. (2010), *The Virtual Theatre District of Pompei.* Available at: http://www.academia.edu/3119884/The_Virtual_Theatre_District_of_Pompeii (Accesssed: 24 January 2015).

Wessels S., Ruther H., Bhurtha R., Schröder R. (2014) ‗Design and creation of a 3D Virtual Tour of the world heritage site of Petra, Jordan', *Proceedings of AfricaGeo*, 1-3 July 2014.

Wikipedia. (2015e) *Cave automatic virtual environment* . Available at: http://en.wikipedia.org/wiki/Cave_automatic_virtual_environment (Accessed: 24 January 2015).

Zamani Project. (2015), Zamani Project, Cape Town, Available at: www.zamaniproject.org/tl_files/data/panotours/Petra/TourWeaver_Petra_Map_Tour.html (Accessed: 24 January 2015).

Zamani Project. (2015), Zamani Project, Cape Town. Available at: www.zamaniproject.org (Accessed: 24 January 2015).

Appendix A

A. Program instructions

A.1 System requirements

The implementation was designed and tested on a PC of the following specifications:

- Windows 7 64-bit
- Intel® Core™ 2 Quad CPU Q9505 @2.83GHz (4 CPUs), ~2.8GHz
- 16 GB system RAM
- NVIDIA GeForce GT 240

The recommended minimum requirements for running the VW application are approximately:

- Windows 7 64-bit
- 2.0+ GHz processor
- 4 GB system RAM
- NVIDIA GeForce GT 240 graphics card or similar.

Please note a proper computer mouse is required for navigation in the VW. A laptop touchpad is not suitable.

A.2 Executable stand-alone program

To run the VW the stand-alone file ―Petra Virtual World.exe" must be executed. No installation is necessary. The appropriate screen resolution must be set for the user‗s computer screen before the VW loads. The other configuration fields, besides screen resolution, should not need to be changed from their default values. Loading of the VW may take several minutes. The escape key on the keyboard exits the application.

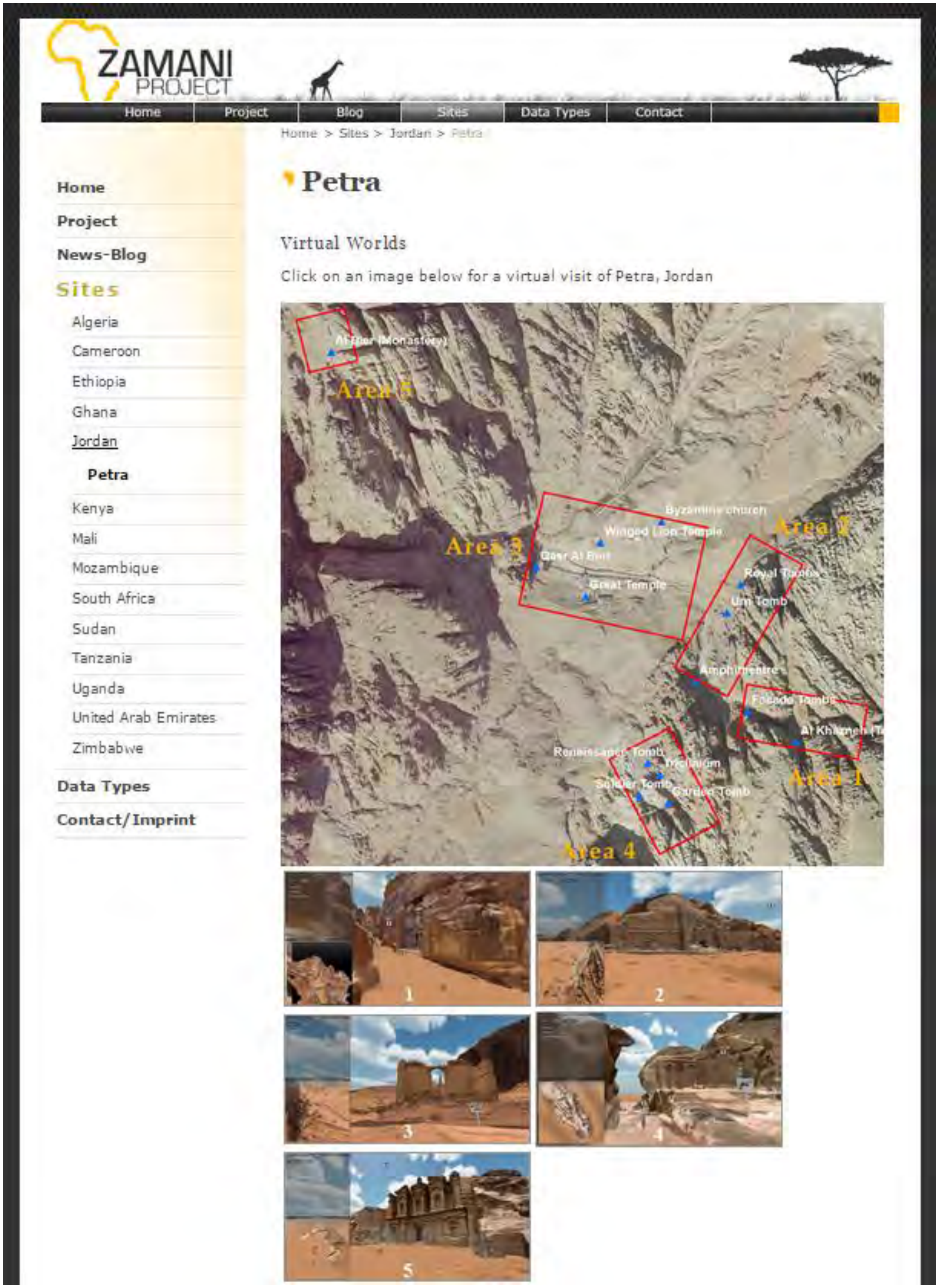
A.3 Online web browser program

The VW of Petra can be accessed online from the following link on the Zamani project website:

http://zamaniproject.org/index.php/petra.html

The five individual scenes have been separated into five Unity web player files and can be seen on the webpage as five screenshot images of each of the scenes (see figure A.1). The reason for not having a single executable Unity web player file to load all the scenes, as is the case of the stand-alone application, was to reduce the amount of data that must be downloaded to run the VW online.

The start screen was omitted from the online VW and in its place the scene layout map has been placed on the webpage to assist users in choosing which scene they wish to run. This can be seen in figure A.1. The user will need to install the Unity Web player to run the VW in their browser. The installation will be initiated automatically when the user clicks on one of the five VW scenes images. Each scene may take several minutes to load.



Petra

Virtual Worlds

Click on an image below for a virtual visit of Petra, Jordan

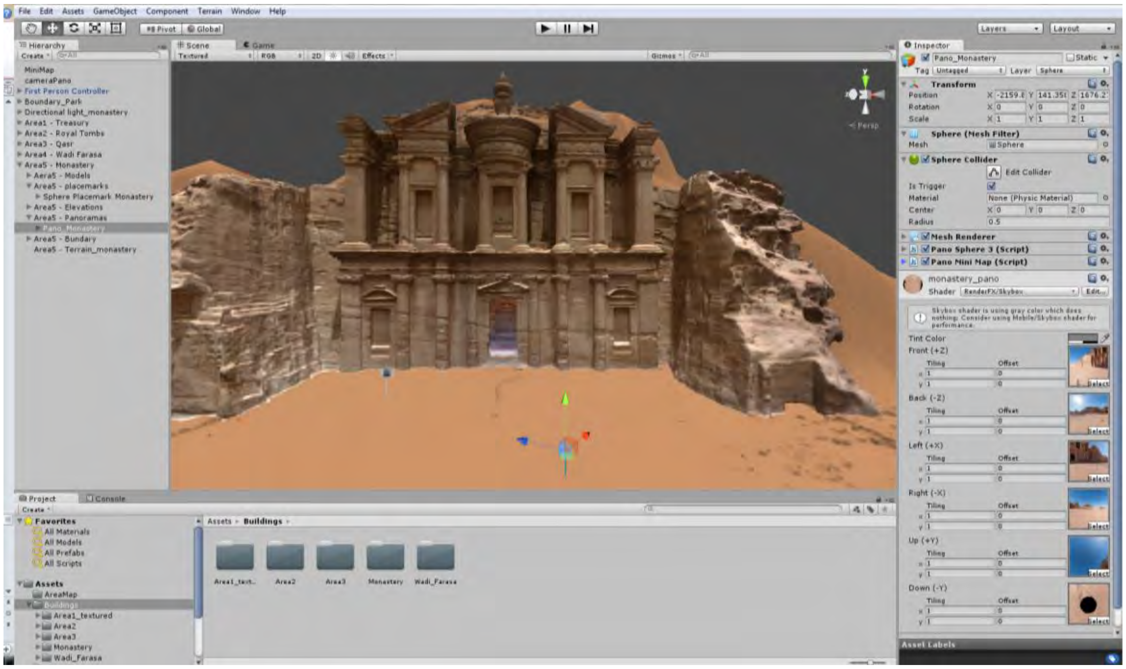
Figure A.1: Virtual World of Petra on the Zamani project website (Zamani Project, 2015).

Appendix B

B. Using Unity to create the Petra VW

B.1 Using Unity

Once the Unity software package is downloaded and installed the developer is presented with the GUI interface as shown in figure B.1. On the left hand side of the interface the hierarchy window shows a list of all the objects that have been imported into the Unity virtual environment. The main screen scene and game viewer shows the virtual environment where the objects are positioned within the scene. The right hand inspector menu shows the properties of imported objects. Colour, size, shading, and script options are manipulated in these menus. The bottom left panel shows the project folder structure. The bottom panel shows the individual objects in their respective folders. Objects can be dragged and dropped into the hierarchy window and then manually positioned within the scene.



☑ Pano_Monastery ☐ Static
Tag Untagged Layer Sphere
Transform
Position X -2159.8 Y 141.35 Z 1676.2
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Sphere (Mesh Filter)
Mesh Sphere
☑ Sphere Collider
Edit Collider
Is Trigger ☑
Material None (Physic Material)
Center X 0 Y 0 Z 0
Radius 0.5
☑ Mesh Renderer
☑ Pano Sphere 3 (Script)
☑ Pano Mini Map (Script)
monastery_pano
Shader RenderFX/Skybox
Skybox shader is using gray color which does nothing. Consider using Mobile/Skybox shader for performance.
Tint Color
Front (+Z) Tiling x 1 y 1 Offset 0 0
Back (-Z) Tiling x 1 y 1 Offset 0 0
Left (+X) Tiling x 1 y 1 Offset 0 0
Right (-X) Tiling x 1 y 1 Offset 0 0
Up (+Y) Tiling x 1 y 1 Offset 0 0
Down (-Y) Tiling x 1 y 1 Offset 0 0

Figure B.1: Screenshot of Unity software.

The ‗play' button at the top of the screen enables the developer to run the application instantly in real-time to view the results of modifications made to scripts and to the virtual environment. This was found to be a very useful tool to continuously evaluate the updates to the VW. During the development of the VW it became obvious that the more detail (high polygon count models and high resolution images) imported into the VW, the more processing power is needed by the end users' computer to run the application smoothly. Thus a compromise had to be found between high detail and efficient running speeds.

B.1.1 Objects

Virtual objects such as terrain, 3D models and virtual characters form the basis of the virtual environment. Objects such as cameras and lighting, which also exist in the environment, are not made visible to the user.

B.1.2 Scripts

To give life to objects in the virtual environment, such as the character that will be virtually exploring the site, scripts are attached to individual objects. Scripts are pieces of code written in Java or C+ that controls the behavioural aspects of objects within the VW. The scripts recognise input from the user, in the form of keyboard or mouse inputs, and interpret this input to instruct the objects to behave in the manner determined by the scripts. It is by creating scripts and their subsequent attachment to objects in the scene that the interaction of the virtual visitor with the virtual environment is designed and developed to the specifications of the developer. Scripts are also able to interact with each other so that input received by one script can effect other scripts.

B.1.3 Scenes

In Unity virtual environments are arranged into scenes. A scene can be thought of as a level in a game. Each scene is a separate entity and can have unique terrain, object and scripting features. Scenes make it easier to control the entire virtual environment by breaking the environment up into geographically grouped objects. Also by splitting the entire environment up into scenes the time taken to load the virtual environment decreases.

B.1.4 Cameras

Camera objects are placed in the scene to capture the virtual environment from a unique view and display this view on the user‘s computer screen. Each scene can include multiple cameras that render different parts of the scene, e.g. a top-view camera shows a top-view orthogonal perspective of the scene whereas a camera attached to the first-person controller shows what the controller would be seeing in the scene. Both cameras display their captured areas simultaneously on the screen in different parts of the screen. For example the top-view camera can be set to display its captured view in a bottom corner of the screen while the first-person controller camera displays its captured view on the rest of the screen. The camera depth hierarchy, meaning which camera has display priority on the screen, is controlled in the cameras inspector menu. The position of the camera‘s display on the screen is also controlled in the inspector. For example the mini-map camera can be chosen to display in the bottom left of the screen and display on top of the first-person controller camera by setting its depth to be above all other cameras.

B.1.5 Unity GUI

To place GUI (graphical user interface) elements, in the form of buttons, windows, menus, images and other elements, into the virtual environment UnityGUI is available as part of Unity‘s scripting application programming interface (API) (Unity3D, 2014).

B.1.6 Lighting

Illumination of the virtual environment is by way of lighting elements placed in each scene. The most common lighting element used is a directional light that represents the sun. Directional lights are placed at an infinite distance and affect all objects in the scene. In this way the terrain and objects are illuminated and shadows are cast onto the scene. Other lighting elements are also available and consist of point lights which shine in all directions, such as a light bulb, spot lights which shine in one direction only, and area lights which shine in all directions from a plane. Additionally ambient light is provided by Unity and can be increased or decreased in intensity (Unity 3D, 2014).

B.2 Importing data into the VW

B.2.1 Importing the terrain into Unity

―Heightmaps" are used in Unity to represent terrain. These are grey-scale images where the pixel intensity value represents a height value. The Unity software has a built-in terrain editor where the extent, shape and texture of a terrain can be artificially created and manipulated.

To import the merged Petra terrain, which is in a mesh format, a tool had to be found to convert the mesh format to a terrain heightmap. Such a tool was available from the Unity community

(Haines, 2012). The resolution for importing the DTM into Unity had to be a compromise between extent of the terrain and displayed detail. A limiting factor is that Unity can only import a meshed model of a maximum of 65000 triangles into the terrain converter. The terrain had to thus be decimated to this number before it was imported. The result of this decimated terrain model was a terrain with approximately 4m resolution covering an area of 1.1km x 1.1km.

B.2.2 Importing the orthophoto into Unity

The orthophoto was imported into Unity by applying it as a texture to the terrain in Unity‘s terrain editor. Options for applying texture to terrain include painting a sample texture on the terrain, or setting a repeated sample texture to cover the terrain. The orthophoto was thus applied to the terrain as a texture, repeated only once.

B.2.3 Importing the 3D models into Unity

3D models are required to be imported in .obj format (mesh format) in Unity with a polygon count of less than 65 000 polygons. If the model exceeds this count it is automatically split into segments of 65 000 polygons or less on import. Unity uses a left hand co-ordinate with the y-axis pointing upwards. Models imported into Unity were therefore transformed where necessary to align with Unity‘s co-ordinate system axis. As mentioned previously, all the models were geo-referenced to the UTM 36N co-ordinate system and when imported into the virtual environment are thus automatically placed into their correct positions with respect to each other. This can be seen in figure B.2.

Figure B.2: Example of 3D models automatic placement in Unity if models have been georeferenced to a common co-ordinate system.

Where the 3D model meets the terrain in the VW the terrain was manually modified by increasing, decreasing or smoothing the terrains geometry using the Unity terrain editor. This was done to create the appearance of a seamless merge between the monument models and the terrain of the surrounding landscape.

B.2.4 Importing the panoramas into Unity

The panoramas were incorporated into the VW by means of the skybox functionality in Unity which uses a full-dome panorama image to display sky in the virtual environment. These skyboxes form the backdrop of the virtual environment scene with all physical objects, including terrain and 3D models, being displayed in front of the skybox. The method of incorporating the Petra panoramas into the VW is explained in section B.3.5.3 .

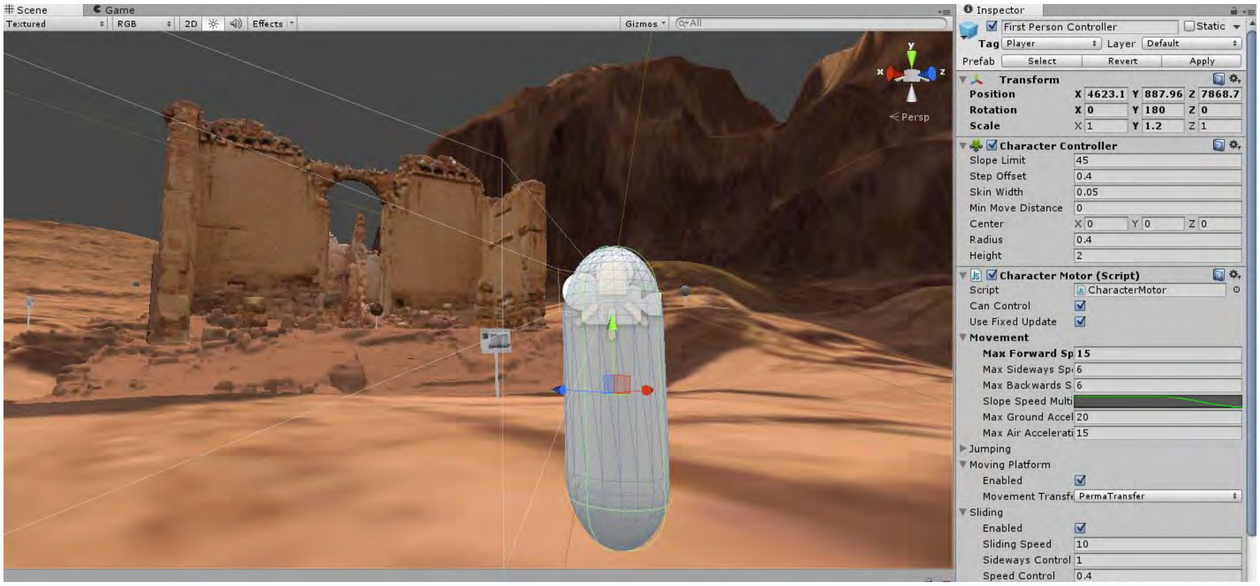
B.3 Creation of the design elements

B.3.1 Start screen

A window is a GUI script function that allows for easy grouping and positioning of GUI elements, and is defined by a border and a text description at the top of the window. The start menu window was thus created to contain all the GUI elements that can be seen in the screen. The window was chosen to display in the centre of the user‘s computer screen and scales in size according to the aspect ratio of the users screen.

B.3.2 First person controller

The standard FPC provided by Unity consists of a physical object, in the form of a capsule with built-in collision detection, and a camera which is located on top of the capsule that captures the scene from the perspective of the FPCs apparent eye-level. The capsule therefore cannot be seen in the first person perspective view of the virtual environment as it does not enter the field of view of the camera. Attached to the FPC are scripts that control the FPC based on keyboard and mouse inputs from the virtual visitor. As can be seen in figure B.3, the properties of the FPC, such as the _walking‘ speed of the FPC and the size and shape of the capsule, can be manipulated in the inspector window in Unity. The FPC capsule was subsequently scaled to be the height of an average person, 1.7m. This was to ensure that when exploring the VW the virtual visitor is seeing the environment at the same eye-level as would be seen in real life. Additionally, when entering rooms and caverns the FPC will only be able to enter if the height of the ceiling is greater than the height of the capsule.



First Person Controller ☐ Static
Tag Player Layer Default
Prefab Select Revert Apply

Transform
Position X 4623.1 Y 887.96 Z 7868.7
Rotation X 0 Y 180 Z 0
Scale X 1 Y 1.2 Z 1

☑ **Character Controller**
Slope Limit 45
Step Offset 0.4
Skin Width 0.05
Min Move Distance 0
Center X 0 Y 0 Z 0
Radius 0.4
Height 2

☑ **Character Motor (Script)**
Script CharacterMotor
Can Control ☑
Use Fixed Update ☑
Movement
Max Forward Sp 15
Max Sideways Sp 6
Max Backwards S 6
Slope Speed Multi
Max Ground Accel 20
Max Air Accelerati 15
Jumping
Moving Platform
Enabled ☑
Movement Transf PermaTransfer
Sliding
Enabled ☑
Sliding Speed 10
Sideways Control 1
Speed Control 0.4

Figure B.3: FPC in the virtual environment in Unity.

B.3.3 Navigation

The ―mouse look” script, supplied by Unity, was modified to create the desired functionality of using the mouse to change the view direction in the virtual environment. In this standard script, which is attached to the FPC camera, the view direction of the FPC‘s camera is controlled by the position of the mouse curser on the screen. When the user moves the mouse the view direction automatically follows the curser. This was found to be impractical and frustrating. If the user wishes to click on the options buttons, on a floating icon, or on the mini-map, the onscreen view direction would follow the mouse curser as the curser moved toward these features to be clicked. To rectify this effect a modification to the ―mouselook” script was necessary.

While the use of the mouse to change the cameras view direction, as opposed to using the arrow keys on the keyboard, was adopted, the unwanted movement associated with moving the mouse curser towards clickable features needed to be removed. To achieve this a modified script, ―mouselook2” was created, based on the code of the standard ―mouselook” script. In this script the camera view is only instructed to move when the user is holding down the left hand mouse button and the mouse is moved across the screen. The curser can then move freely across the screen without affecting the camera view as long as the left hand mouse button is not held down.

Additionally, when the curser moves over areas two and three of the screen, namely the mini-map and menu options box, and depending if these windows are active (not minimised), the camera view movement is automatically disabled even if the user is holding down the left hand mouse button. Code for these modifications can be seen in figure B.4 below.

```
if(Input.GetMouseButton(0)) // the mouse is only active when the left mouse button is held down
    {
    if (map==true && options ==true){ // if the options window and minimap are both not minimised

        if (Input.mousePosition.x < Screen.height/2)
        {
        MouseActive=false; //de-activate mouse input
        }
        else
        {
        MouseActive=true; //activate mouse input
        }
    }
    if (map == true && options == false) // if the options is minimised but minimap is not minimised
    {
        if (Input.mousePosition.x < Screen.height/2  && Input.mousePosition.y < Screen.height/2)
        {
        MouseActive=false; //de-activate mouse input
        }
        else
        {
        MouseActive=true; //activate mouse input
        }
    }
    if (map == false && options == false) //if  the options window and minimap are both minimised
    {
    MouseActive=true;
    }

    if (map == false && options == true) //if the options is not minimised but minimap is minimised
    {
        if (Input.mousePosition.x < Screen.height/2  && Input.mousePosition.y > Screen.height/2)
        {
        MouseActive=false; //de-activate mouse input
        }
        else
        {
        MouseActive=true; //activate mouse input
        }
    }
```

Figure B.4: The ―mouselook2” script code that enables camera movement only when the left-hand mouse button is held down. The camera view movement is also disabled when the mouse curser is over areas two and three (see figure 5.20) and these windows are not minimised. The code that was added to the script to enforce these stipulations is shown here.

This script was then attached to the camera connected to the FPC and also to the camera that displays the Petra panoramas, discussed in B.3.4.1.

The second navigation modification was to the input received by the ―FPC input controller” script. This script receives input from the keyboard and instructs the FPC to which the script is attached to act upon this input. In its standard form the arrow buttons on the keyboard control the movement of the FPC. The forward arrow button moves the FPC forward, the left and right buttons move the character sideways, but keep the view direction of the FPC facing forward, and the down button moves the FPC backwards in a circular motion. It was decided that only the forward button was necessary as the other three buttons introduce confusion to the navigation. These sideways and down-arrow buttons were subsequently disabled in the Unity project settings. Additionally the ―W” button on the keyboard was assigned to the same function as the up arrow. This was based on ergonomic considerations with respect to the handling of keyboard and mouse. The user‘s left hand would typically make use of the keyboard which controls forward motion, while the right hand would handle the mouse. The ―W” button is closer to the user‘s left hand then the up arrow, and is more convenient to access on the keyboard. The spacebar button, which causes the virtual visitor to jump onto objects or climb stairs, is then easily accessed by the thumb of the user‘s left hand.

B.3.4 Pop-up information

B.3.4.1 Panoramas

The position of each panorama was established by visual inspection of the panorama and the virtual environment with the proximity of the panorama to objects such as 3D models and rocks on the ground aiding this placement.

By placing the edge of the panorama image in line with the z-axis of the Unity co-ordinate system the alignment of the panorama could be matched to the virtual environment. To determine this alignment the position of the panorama in the virtual scene was navigated to and the viewing direction of the camera was directed along the z-axis of the Unity co-ordinate system. The feature of the Petra landscape or monument that is then visible at the centre of the virtual scene should be the same as what is visible at the edge in the panorama. The centre of panorama was then subsequently shifted, in image manipulation software, so that its edge aligned visually with the scenes centre. This method can be seen in figure B.5 below.

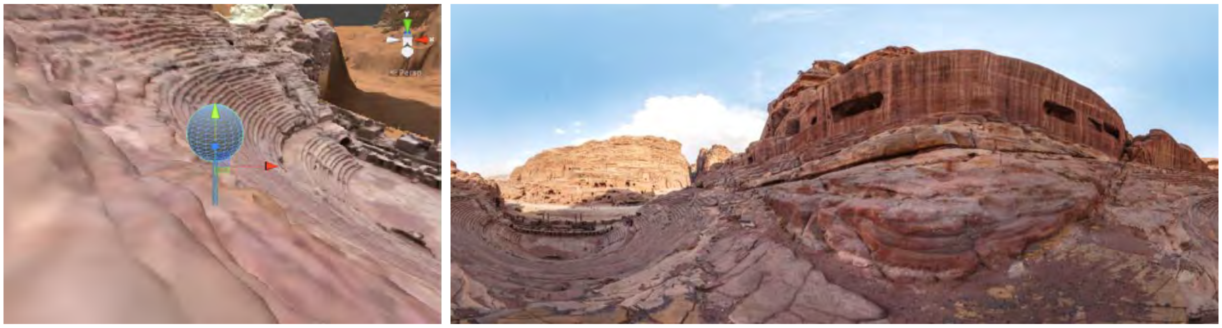


Figure B.5: Method of panorama image alignment in Unity. Right: Panorama with edge aligned to the co-ordinate axis of Unity.

Once aligned, the panoramas were then converted into a Unity skybox to be displayable as 360-degree full-dome images. The panorama object as seen in the virtual environment (figure 5.22) is made up of the following components: a thin cylinder, a transparent sphere that sits on top of the cylinder and acts as a place-mark for the hovering panorama icon, and a script that is attached to the sphere that draws the hovering clickable icon at the spheres centre and activates the panorama skybox when clicked.

The first task of the script is to display the hovering icon above the panorama cylinder at the centre of the transparent sphere. The icon is in the form of a GUI button and is represented by a blue dot. The position of the icon on the screen is calculated by converting the panorama sphere 3D virtual world co-ordinates into 2D screen co-ordinates. The icon remains the same size on the screen no matter the distance between the virtual visitor and the panorama due to it being a GUI button element.

The icon should only be made visible in the virtual environment if the virtual visitor has a direct line of site to the panorama object with no other objects obstructing the view. To achieve this constraint the script casts a ray from the FPC camera to the centre of the sphere and if the ray returns true, i.e. there are no obstructions, the icon is displayed. To prevent the screen becoming cluttered with panorama icons the icons were set to only be displayed on screen if the distance between the visitor and the panorama is less than 120m. The code for these functions can be seen in figure B.6 below.

```
    var hit : RaycastHit;
    var rayDirection = maincammi.transform.position - transform.position;
    var allButTriggers = ~(1 << 10); //  only check ray cast shot to the objects labeled sphere
    if (Physics.Raycast (transform.position, rayDirection, hit, 100000, allButTriggers))
    {
        if (hit.transform == player)// if the hit is true
        {
        sightline = true;   // panorama object can see the player
        }
        else
        {
        sightline = false; //there is something obstructing the view
        }
    }
    if (Input.GetKeyDown("return") || Input.GetKeyDown("w")|| Input.GetKeyDown("up")) //returns to FPC camera view
    {
    PanoCam.depth = 0; // puts the panocam underneath all other cameras
    }
}

function OnGUI()
{
    //displaying the icon
    var point = Camera.main.WorldToScreenPoint(sphere.position); // covert from world to screen points to display the icon
    rect.x = point.x -14; // shift the icon so its in the centre
    rect.y = Screen.height - point.y-14; // bottom left corner set to the 3D point.
    var off = Vector3.zero;
    off = Camera.main.WorldToViewportPoint(sphere.position);
    if (window_options.map == true) // if the minimap is on then dont draw icons in the map area
    {
    red = window_options.absolouteWidth;// the screen co-ords where the icon can be drawn
    fed = Screen.height/2;// the screen co-ords where the icon can be drawn
    }
    if (window_options.map == false) // if the minimap is on then dont draw icons in the map area
    {
    red = 0;
    fed = 10000;
    }
    var distance = Vector3.Distance(player.transform.position, sphere.transform.position);
    if  (rect.x < red && rect.y > fed || distance > 120) // do nothing if the icon would be drawn in the minimap,
                                                        //or the distance between pano and player is more than 50m

    {
    //do nothing
    }
    else
    {
```

Figure B.6: Code to draw the panorama icon if there is no obstruction to line of site and the distance between FPC and panorama object is less than 120m.

When the visitor clicks on the panorama icon the script instructs the FPC (virtual visitor) to move to the location of the panorama in the scene. A camera, exclusively dedicated to the display of panorama skyboxes, then takes the places of the FPC camera in the camera depth hierarchy. The chosen panorama skybox is then enabled, giving the visitor a 360-degree photographic view of the scene. All objects in the virtual environment are made invisible to the user, so that the user experiences the skybox as a full 360-degree panoramic view. The only objects that are left visible in the scene are the other hovering panorama icons, which the user can directly click on to jump to the next panorama.

B.3.4.2 Elevations, sections and plans

To achieve pop-up data functionality for the sections, elevations and plans a script was created and attached to the virtual signboard (see figure 5.24). The first function of the script is to determine if there is any object obstructing the line of site between visitor and signboard. The method is the same as used for the panorama icons, by casting a ray between the FPC camera object and the centre of a transparent sphere situated on top of the signboard. If the view is unobstructed the green star icon is displayed hovering over the signboard at the position of the centre of the sphere. Additionally, if the distance between the user and the signboard is more than 120m the icon will not be displayed.

When the hovering green star icon is clicked the script calls up a GUI window containing the image of the section, elevation or plan. The image is displayed in full-screen mode and is scaled to its correct dimensions on the screen, depending on the users individual screen dimensions. While the image is displayed in full-screen mode all hovering icons, such as other signboard icons and panorama icons, are turned off so that they do not interfere with the view of the image.

It should be noted that any image, such as a photograph, map, historical text etc., can be displayed by the signboard by simply substituting the image.

B.3.4.3 Raised viewing platforms/3D model placemarks

To achieve the raised viewing platform functionality, a flat, invisible cylinder platform object was created and placed at a chosen location offering an elevated view of a monument in the VW. A script was coded to display the up-arrow icon and facilitate transportation of the FPC to this location. The script was then attached to the platform object. As with the panorama and pop-up data script, unobstructed line of site must first be attained before placing the icon on the screen. When the user clicks the icon the scripts instructs the FCP position to be relocated to the top of the invisible ledge.

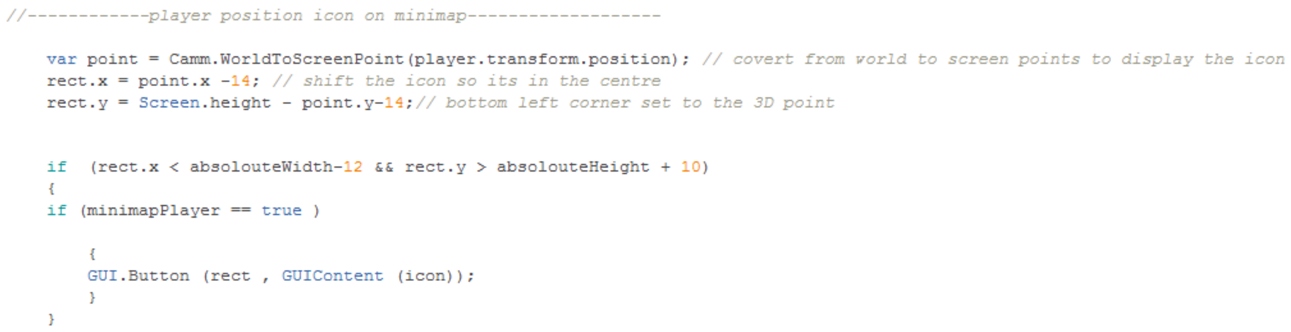
B.3.5 The mini-map

B.3.5.1 The mini-map window

To display the mini-map on the screen a dedicated mini-map camera object was created in Unity and placed at the top of the camera depth hierarchy so that it displays above all other cameras on the screen. The camera was positioned in the scene above the virtual environment pointing downwards to capture the entire extent of the VW. The camera was set to display an orthographic projection of its captured area in the bottom left of the screen. The shape and size of the mini-map was set to be a square with the length of the side of the square set to half the entire screen height. In this way the mini-map scales correctly to each user‘s individual screen dimensions. A GUI window was then created by a script attached to the mini-map camera to draw a border around the mini-map and display buttons on the mini-map.

B.3.5.2 Icon following user

To position the icon on the mini-map the 3D world co-ordinates of the visitor in the virtual environment are transferred to the mini-map cameras 2D screen co-ordinates as can be seen in the figure B.7.. As the visitor explores the scene the position of the mini-map camera is set to that of the visitor. In this way the user is always centrally located on the mini-map



```
//------------player position icon on minimap-------------------

    var point = Camm.WorldToScreenPoint(player.transform.position); // covert from world to screen points to display the icon
    rect.x = point.x -14; // shift the icon so its in the centre
    rect.y = Screen.height - point.y-14;// bottom left corner set to the 3D point


    if  (rect.x < absolouteWidth-12 && rect.y > absolouteHeight + 10)
    {
    if (minimapPlayer == true )

        {
        GUI.Button (rect , GUIContent (icon));
        }
    }
```

Figure B.7: Script code converting 3D world co-ordinates of the FPC to 2D screen co-ordinates of the mini-map.

B.3.5.3 Panorama icons

To display the panorama icons on the mini-map a script was created and attached to each panorama object in the scene. The script converts the position of the panorama from 3D world co-ordinates in the virtual environment to 2D screen co-ordinates on the mini-map display. When the visitor clicks on the panorama the user is transported to the chosen panorama‘s location and the panorama is activated.

B.3.5.4 Raised viewing platform icons/3D model placemarks

The raised viewing platform icons on the mini-map are displayed via a script that was created and attached to each viewing platform object in the virtual environment. The script acts in the same way as the panorama icons script described above and converts the position of the raised viewing platform from 3D world co-ordinates in the virtual environment to 2D screen co-ordinates on the mini-map display. The user is transported to the chosen raised viewing platform when the visitor clicks on the platform icon. The icons have tooltips associated with them so that they also act as placemarks for the monuments in the mini-map.

B.3.5.5 Zooming and panning

To zoom in and out of the mini-map two methods were employed. In the first method two buttons were created and positioned at the top left corner of the mini-map. These buttons are indicated with –+” and –-–signs and have attached tooltips which read –zoom in” and –zoom out” respectively. When a user clicks on these buttons the size of the area that the mini-map is currently showing either expands or contracts. Limits were set on the extent of the zoom so that it is not possible to zoom in or out past certain predefined levels. This safeguards against the user zooming in through the mini-map or so far out that the mini-map is not recognisable. The same zooming function was then added to the scroll wheel on the mouse so that when the mouse icon is above the mini-map and the scroll is activated the size of the map increases or decreases. This was achieved with the code shown in figure B.8 below.

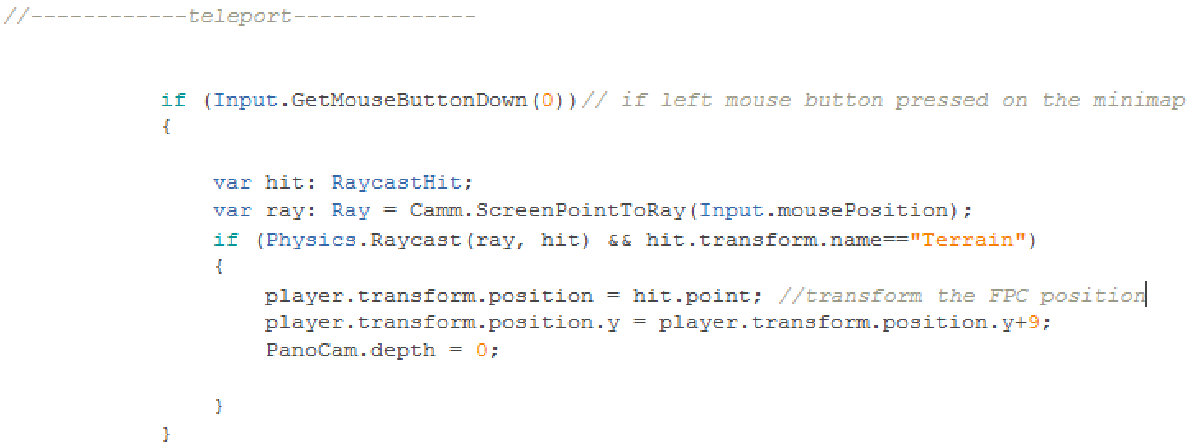
```
//------------------------zoom in MINI MAP------------

    if (GUI.Button (Rect (10, 3, 20, 20 ), GUIContent("+","Zoom In")))
    {

        if (Camm.orthographicSize -zIncrements >0) // to enusre you dont zoom in past the terrain
        {
        Camm.orthographicSize = Camm.orthographicSize -zIncrements ; // change the orthographic size of the camera
        }
    }

    if(GUI.tooltip =="Zoom In")
    {
    GUI.Label(Rect(10, 20, 100, 20), GUI.tooltip);
    }

//----------------------zoom out MINI MAP--------------

    if (GUI.Button (Rect ( 30, 3, 20, 20),GUIContent( "-", "Zoom Out")))
    {
        if ( Camm.orthographicSize < zoom -0.01) //increasing orthographicSize zooms out, zoom = 612.776
        {
        Camm.orthographicSize = Camm.orthographicSize + zIncrements; // change the orthographic size of the camera
        }
    }
    if(GUI.tooltip =="Zoom Out")
    {
    GUI.Label(Rect(30, 20, 100, 20), GUI.tooltip);
    }
```

Figure B.8: Script code showing zooming in the mini-map.

To pan the visitor must click on the mini-map with the right mouse button and hold this button down while moving the mouse. When the right mouse button is let go the panning functionality de-activates. This action serves to drag the mini-map to the desired location.

B.3.5.6 Teleporting

To jump to a desired location in the VW the user is able to click on the mini-map to be teleported to this clicked location. This functionality is part of the mini-map script and only works if the user clicks on the terrain orthophoto in the mini-map and not the 3D models. Teleporting is achieved by the raycast function, which casts a ray to the terrain, and coverts the hit point, i.e. where the user has clicked on the terrain in the mini-map, into 3D world co-ordinates. The visitor is then transported to this location. The code for this can be seen in figure B.9 below.



```
//------------teleport--------------


            if (Input.GetMouseButtonDown(0))// if left mouse button pressed on the minimap
            {

                var hit: RaycastHit;
                var ray: Ray = Camm.ScreenPointToRay(Input.mousePosition);
                if (Physics.Raycast(ray, hit) && hit.transform.name=="Terrain")
                {
                    player.transform.position = hit.point; //transform the FPC position
                    player.transform.position.y = player.transform.position.y+9;
                    PanoCam.depth = 0;

                }
            }
```

Figure B.9: Teleporting script code.